

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет міжнародних відносин
Кафедра комп'ютерних мультимедійних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ С.М. Лобода

“ ____ ” _____ 2020 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР
ЗА СПЕЦІАЛІЗАЦІЄЮ „ТЕХНОЛОГІЇ ЕЛЕКТРОННИХ
МУЛЬТИМЕДІЙНИХ ВИДАНЬ”

Тема: „Технологія створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”

Виконавець: студент 215М групи Губар Олексій Вячеславович

Керівник: доктор педагогічних наук, професор Лобода Світлана Миколаївна

Нормоконтролер: _____ ст. викладач Таран Віктор Миколайович

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет міжнародних відносин

Кафедра комп'ютерних мультимедійних технологій

Спеціальність: 186 „Видавництво та поліграфія”

Освітньо-професійна програма: „Технології електронних мультимедійних видань”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ С.М. Лобода

" ____ " _____ 2019 р.

ЗАВДАННЯ

на виконання дипломної роботи

Губаря Олексія Вячеславовича

1. Тема дипломної роботи „Технологія створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова” затверджена наказом ректора від 02.10.2019 р. №2255/ст.

2. Термін виконання роботи: з 7 жовтня 2019 року по 28 січня 2020 року.

3. Вихідні дані до роботи: текстові матеріали для наповнення веб-ресурсу кафедри, ілюстративний матеріал.

4. Зміст пояснювальної записки: теоретичні засади створення освітніх веб-ресурсів. Порівняльний аналіз технологій створення веб-ресурсу. Проектна реалізація технології створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

5. Перелік обов'язкового ілюстративного матеріалу: рисунки, таблиці, презентація *PowerPoint*.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1	Отримання завдання на виконання дипломної роботи	07.10.2019	
2	Підбір та вивчення наукової літератури за темою дипломної роботи	08.10.2019-05.11.2019	
3	Аналіз існуючих веб-ресурсів кафедр	06.11.2019-15.11.2019	
4	Створення концепції та макету веб-ресурсу кафедри, вибір програмного забезпечення та <i>CMS</i>	16.11.2019-30.11.2019	
5	Проектна реалізація технології створення веб-ресурсу	01.12.2019-31.12.2019	
6	Оформлення пояснювальної записки дипломної роботи	01.01.2020-22.01.2020	
7	Проходження нормоконтролю	23.01.2020	
8	Підготовка презентації та доповіді	24.01.2020-26.01.2020	
9	Отримання відгуку керівника, рецензії	27.01.2020	

7. Дата видачі завдання “7” жовтня 2019 р.

Керівник дипломної роботи: _____ Лобода С.М.

Завдання прийняв до виконання: _____ Губар О.В.

РЕФЕРАТ

Пояснювальна записка до дипломної роботи „Технологія створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”: 107 сторінок, 63 рисунка, 3 таблиці, 73 використаних джерел, 2 додатки.

ВЕБ-РЕСУРС, КЛАСИФІКАЦІЯ ВЕБ-РЕСУРСІВ, ПРИНЦИПИ ПРОЕКТУВАННЯ ВЕБ-РЕСУРСІВ, *FRONT-END*, *BACK-END*, СИСТЕМА КЕРУВАННЯ ВМІСТОМ, ЕТАПИ ТЕХНОЛОГІЇ СТВОРЕННЯ ВЕБ-РЕСУРСУ, *SEO*-ОПТИМІЗАЦІЯ, АДАПТИВНІСТЬ, КРОСПЛАТФОРМЕНІСТЬ.

Об’єктом дослідження є веб-ресурс, як інформаційно-комунікаційний засіб.

Предмет дослідження – технологія створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

Мета дипломної роботи – розробити технологію створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

Методи дослідження: теоретичні – аналіз, порівняння, систематизація та узагальнення наукової літератури вітчизняних і зарубіжних авторів, електронних ресурсів для з’ясування стану розробленості проблеми технології створення веб-ресурсу; методи системного аналізу для визначення видо-типологічної класифікації та принципів проектування веб-ресурсів; порівняльний аналіз для вибору програмного забезпечення та системи керування веб-ресурсом; синтез для розробки інформаційно-концептуальної моделі веб-ресурсу; емпіричні – експеримент для виявлення оптимальних налаштувань *SEO*-оптимізації; тестування для аналізу зручності користування веб-ресурсом.

Технічні та програмні засоби: *CorelDRAW*, *Sublime Text 3*, *CMS WordPress*.

Рекомендації щодо використання результатів: розроблену технологію створення веб-ресурсів можуть використовувати веб-розробники, як методичні рекомендації для створення ресурсів відповідного спрямування.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	6
ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ОСВІТНІХ ВЕБ-РЕСУРСІВ	10
1.1. Видо-типологічна класифікація веб-ресурсів	10
1.2. Принципи проектування веб-ресурсів	20
1.3. Порівняльний аналіз веб-сторінок структурних навчальних підрозділів закладів вищої освіти	29
Висновки до розділу	39
РОЗДІЛ 2. ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ ВЕБ-РЕСУРСУ	40
2.1. Порівняльний аналіз програмного забезпечення.....	40
2.2. Структура взаємодії <i>front-end</i> та <i>back-end</i>	49
2.3. Вибір системи керування веб-ресурсом	57
2.4. Етапи технології створення веб-ресурсу	64
Висновки до розділу	69
РОЗДІЛ 3. ПРОЕКТНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЙ СТВОРЕННЯ ВЕБ-РЕСУРСУ „КАФЕДРА СУРДОПЕДАГОГІКИ ТА СУРДОПСИХОЛОГІЇ ІМЕНІ М.Д. ЯРМАЧЕНКА НАЦІОНАЛЬНОГО ПЕДАГОГІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ М.П. ДРАГОМАНОВА”	70
3.1. Інформаційно-концептуальна модель веб-ресурсу	70
3.2. Написання програмного коду та верстка веб-ресурсу	78
3.3. Тестування веб-ресурсу	88
Висновки до розділу	93
ВИСНОВКИ.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	98
Додаток А	106
Додаток Б.....	107

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Веб-ресурс	місце розташування, що ідентифіковане у вигляді <i>URL</i> -адреси в мережі Інтернет, де зберігаються веб-сторінки для доступу та використання.
Плагін	додаток, незалежно скомпільований програмний модуль, що динамічно підключається до основної програми, призначений для розширення або використання її можливостей.
<i>CSS</i>	<i>Cascading Style Sheets</i> – каскадні таблиці стилів.
<i>CMS</i>	<i>Content Management System</i> – система керування вмістом.
<i>HTML</i>	<i>Hypertext Markup Language</i> – мова розмітки гіпертекстових документів.
<i>PHP</i>	<i>Hypertext Preprocessor</i> – гіпертекстовий препроцесор.
<i>SEO</i>	<i>Search Engine Optimization</i> – пошукова оптимізація.
<i>UI</i>	User Interface – інтерфейс користувача.
<i>URL</i>	<i>Uniform Resource Locator</i> – уніфікований показчик ресурсу.
<i>UX</i>	User Expreience – досвід користувача.
<i>WYSIWYG</i>	<i>What You See Is What You Get</i> – що бачиш, то і отримаєш.

ВСТУП

Актуальність теми. В умовах сьогодення актуальною проблемою є комп'ютеризація освіти, яка спрямована на підготовку професіоналів відповідної галузі. Одним з ланцюжків комп'ютеризації освітнього процесу є веб-ресурси закладів вищої освіти та їхніх структурних підрозділів. Вони, в першу чергу, виконують інформативні функції (вступні іспити, розклад занять, літературні джерела за профілем навчання) та комунікативні (обмін досвідом, спільні наукові дослідження), а також дозволяють запровадити проекти, що займаються дистанційним навчанням і допомагають особам, не виходячи з дому, отримувати відповідні знання.

Підходи до створення веб-ресурсів, а саме використання семантичної розробки, відповідність стандартам (специфікаціям) та делегування відповідальності мов описано в наукових працях Д. Котерова [9], Д. Дакетта (*John Duckett*) [6], М. Мак-Дональда (*Matthew MacDonald*) [12, 13], Д. Макфарланда (*David McFarland*) [14], Р. Ніксона (*Robin Nixon*) [19] та інших.

Проблема видо-типологічної класифікації веб-ресурсів розкривається в наукових дослідженнях Т. Самойлової [23, 24], Д. Барраклау (*Dan Barraclough*) [30], Д. Мілецького (*Jason Miletsky*) [55] та інших, яка є важливим фактором для користувачів, оскільки вказує на функціональні можливості та сприяє ефективному пошуку необхідної інформації.

Основні положення принципів проектування, які стосуються оформлення мультимедійного вмісту та *SEO*-оптимізації, котрі необхідно враховувати при створенні веб-ресурсів, розглянуто в наукових працях І. Мальцева [15], Ш. Мура (*Shawn Moore*) [56], Р. Томала (*Rafal Tomal*) [25], К. Чепмен (*Cameron Chapman*) [38] та інших.

Мета дипломної роботи – розробити технологію створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

Відповідно до визначеної мети були сформульовані **основні завдання дипломної роботи:**

1) дослідити видо-типологічну класифікацію та принципи проектування веб-ресурсів;

2) проаналізувати сучасне програмне забезпечення та системи керування веб-ресурсів;

3) описати етапи технології створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”;

4) зробити проектну реалізацію веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

Об’єктом дослідження є веб-ресурс, як інформаційно-комунікаційний засіб.

Предмет дослідження – технологія створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”.

Методи дослідження:

– теоретичні: аналіз, порівняння, систематизація та узагальнення наукової літератури вітчизняних і зарубіжних авторів, електронних ресурсів для з’ясування стану розробленості проблеми технології створення веб-ресурсу; методи системного аналізу для визначення видо-типологічної класифікації та принципів проектування веб-ресурсів; порівняльний аналіз для вибору програмного забезпечення та системи керування веб-ресурсом; синтез для розробки інформаційно-концептуальної моделі веб-ресурсу;

– емпіричні: експеримент для виявлення оптимальних налаштувань *SEO*-оптимізації; тестування для аналізу зручності користування веб-ресурсом.

Наукова новизна дослідження полягає в тому, що:

– *вперше*: створено веб-ресурс кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова;

– *уточнено*: принципи проектування та етапи технології створення веб-ресурсу; інструментарій покращення функціональних можливостей програмного забезпечення та систем керування вмістом;

– *дістали подальшого розвитку*: теоретичні та методичні засади технології створення веб-ресурсів структурних навчальних підрозділів закладів вищої освіти.

Практичне значення отриманих результатів полягає у створенні веб-ресурсу кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка з дотриманням принципів проектування: дизайну, програмування та *SEO*-оптимізації. Результати розробленої технології створення веб-ресурсів для структурних навчальних підрозділів можуть використовуватися веб-розробниками, як методичні рекомендації для створення ресурсів відповідного спрямування. Матеріали дослідження можуть бути використані в навчальному процесі при викладанні дисциплін „Мультиплатформені програмні засоби мультимедіа”, „Гіпертекстова обробка інформації”, а також в процесі самостійного набуття знань.

Результати дослідження впроваджено в інформаційний простір факультету спеціальної та інклюзивної освіти Національного педагогічного університету імені М.П. Драгоманова (довідка №106/2019 від 26.09.2019).

Апробація результатів. Основні положення і результати дипломного дослідження обговорювалися під час виступів на засіданнях кафедри комп’ютерних мультимедійних технологій Національного авіаційного університету та кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ОСВІТНІХ ВЕБ-РЕСУРСІВ

1.1. Видо-типологічна класифікація веб-ресурсів

Проблемі вивчення видо-типологічних класифікацій приділяла увагу значна кількість вітчизняних і зарубіжних дослідників, зокрема, В. Берковський [2], А. Гушев [5], Н. Пасічник [20], Й. Ривкінд [21], С. Матвеєва [52], Т. Самойлова [23, 24], Д. Барраклау (*Dan Barraclough*) [30], Н. Грехем (*Nicole Graham*) [44], М. Кіпп (*Margaret Kipp*) [47], С. Майер (*Sven Meyer zu Eissen*) [54], Д. Мілецький (*Jason Miletsky*) [55], М. Сатія (*M. Satija*) [63], А. Славік (*Aida Slavic*) [67], С. Чандра (*Sarvesh Chandra*) [37] та інші.

Відповідно до Міжнародного стандартного бібліографічного опису для комп'ютерних файлів та електронних ресурсів (*ISBD(ER)*), „веб-ресурс – це місце розташування, що ідентифіковане у вигляді *URL*-адреси в мережі Інтернет, де зберігаються веб-сторінки для доступу та використання” [45, с. 14].

Р. Раджалакшмі (*R. Rajalakshmi*) та С. Аравіндан (*C. Aravindan*) тлумачать класифікацію веб-ресурсів, як процес визначення певних ознак, що базуються на аналізі контенту з метою чіткої та однозначної ідентифікації [61, с. 323]. Класифікація допомагає користувачам конкретизувати пошукові інтереси до тематичних документів у веб-середовищі.

Суть проблеми класифікації веб-ресурсів полягає в тому, що користувачі не в змозі визначити вид певної сторінки. Д. Руссінов (*Dmitry Roussinov*), К. Кроустон (*Kevin Crowston*), М. Нілан (*Mike Nilan*), Б. Квасник (*Barbara Kwasnik*) та Д. Цай (*Jin Cai*) стверджують, що вид можна визначити за призначенням або функцією, з точки зору фізичної форми або в формі документа та, як правило, вид поєднує в собі мету й форму [62, с. 2].

Досліджуючи веб-ресурси за видовими ознаками, Т. Самойлова [23] надає наступну класифікацію веб-ресурсів (рис. 1.1): комерційний, корпоративний та інформаційний.



Рис. 1.1. Класифікація видів веб-ресурсів за Т. Самойловою

Ресурси комерційного типу є найбільш поширеними серед інших видів ресурсів. Основна мета їх створення – продавати товари користувачам мережі. Комерційні ресурси поділяють на кілька основних видів [23]:

- ресурс-візитка (створюється для однієї людини або невеликої фірми; метою створення є поширення професійної інформації про компанію або певного фахівця для великої кількості відвідувачів мережі Інтернет, що сприяє залученню нових клієнтів);

- інтернет-магазин (надає послуги в оформленні купівлі та доставки різноманітних товарів у режимі реального часу (*on-line*); відвідувач може вибрати товар та зробити замовлення, обрати варіант оплати та спосіб доставки замовлення, а також одержати рахунок на оплату);

- промо-ресурси (призначені для активної реклами будь-якого продукту, товару або послуги; вони зазвичай невеликі за об’ємом текстового наповнення, але повинні мати яскравий, незабутній та стильний дизайн, що поєднує велику кількість графіки і здатний ефектно проінформувати користувача про товар чи послугу);

– ресурс-вітрина (містить максимальну інформацію про власника та надає представницький каталог його товарів або послуг; зазвичай містить прайс-листи, описи товарів або послуг, фотогалереї).

Ресурси корпоративної спрямованості створюються для виконання одного з наступних завдань: підвищення іміджу компанії в очах існуючих і потенційних клієнтів або зміцнення зв'язку між відділеннями та філіями організації. Відповідно до цього корпоративні ресурси бувають [23]:

- іміджеві ресурси;
- інформаційні ресурси.

Інформаційні ресурси призначені для донесення до користувачів найрізноманітнішої інформації. Це досить великі віртуальні масиви, що включають в себе безліч тематичних розділів меншого обсягу, або певну кількість самостійних проєктів. Вони можуть бути для відвідувачів значним джерелом інформації, нагадуючи спеціалізований журнал або енциклопедію. Існує кілька різновидів ресурсів інформаційної спрямованості [23]:

- тематичний ресурс;
- ресурс новин;
- форум (служать для забезпечення спілкування всіх бажаючих у форумі);
- блог.

Т. Самойлова [24] наводить типологічну класифікацію веб-ресурсів (рис. 1.2): залежно від мети створення, за ступенем доступу, за функціональністю та стилем оформлення, за фізичним розташуванням та за функціональною спрямованістю.

Залежно від мети створення веб-ресурси поділяються на [24]:

- комерційні (основним призначення є розвиток бізнесу за допомогою засобів інтернет-технологій; вони сприяють отриманню доходів від продажу товарів, пошуку нових клієнтів, розміщенню реклами тощо);
- некомерційні (представництва урядових організацій, освітніх установ, політичних партій; основним завданням є повідомлення про своє існування, надання інформації про діяльність, створення засобів спілкування, надання безкоштовних послуг з певних питань).



Рис. 1.2. Класифікація типів веб-ресурсів за Т. Самойловою

За ступенем доступу веб-ресурси бувають [24]:

- відкриті (доступ має будь-який користувач);
- напіввідкриті (для доступу до ресурсу необхідна реєстрація);
- закриті (право доступу має обмежене коло осіб).

За функціональністю та стилем оформлення веб-ресурси розрізняються [24]:

- динамічні (генерують свій вміст спеціальними програмами, що мають назву скрипти, і формують інформацію в залежності від дій користувача на підставі аналізу його попередніх дій);

– статичні (інформаційне наповнення залишається незмінним для всіх користувачів ресурсом);

– флеш-ресурси (конструюються за допомогою технології використання анімаційних зображень *Adobe Flash*, саме тому вони дуже яскраві та особливо привабливі).

За фізичним розташуванням веб-ресурси поділяються на [24]:

– локальні (доступні тільки для конкретної локальної мережі);

– глобальні (надають доступ для будь-якого користувача мережі).

За функціональною спрямованістю веб-ресурси виокремлюють [24]:

– інтернет-представництва різних організацій і компаній;

– веб-сервіси (служать для надання тих чи інших послуг та завдань);

– інформаційні ресурси (включають в себе текстові, ілюстративні, мультимедійні матеріали або повідомлення інших будь-яких видів, що дозволяють розкрити тематичний зміст ресурсу або одного з його розділів);

– соціальні мережі (інтерактивний веб-ресурс, призначений для можливості надання засобів зв'язку для спілкування з іншими користувачами).

Д. Барраклау (*Dan Barraclough*) [30] класифікує веб-ресурси за такими критеріями, як функціональні можливості, тип наповнення, ступінь оптимізації та призначення (рис. 1.3).

За функціональними можливостями веб-ресурси поділяються на п'ять основних різновидів: брошурний, електронної комерції, портал, *wiki*-ресурс, соціальні медіа [30].

Брошурний веб-ресурс являє собою найпростіший тип ресурсу з точки зору функціональності. Він створюється для одного користувача або невеликої фірми з метою поширення професійної інформації про компанію або певного фахівця для великої кількості відвідувачів мережі Інтернет. Зазвичай такий ресурс включає в себе декілька сторінок, що містять контактні дані для майбутніх клієнтів, історія фірми, заслуги та досягнення, фотографії тощо.

Веб-ресурс електронної комерції представляє собою ресурс, за допомогою якого користувачі можуть сплачувати за товари чи послуги в Інтернеті. Зазвичай це

стосується однієї компанії, що надає послуги великій кількості користувачів, проте це може відноситися до веб-ресурсів, відомих як маркетплейси, що виконують роль торговельних посередників між постачальниками та споживачами.

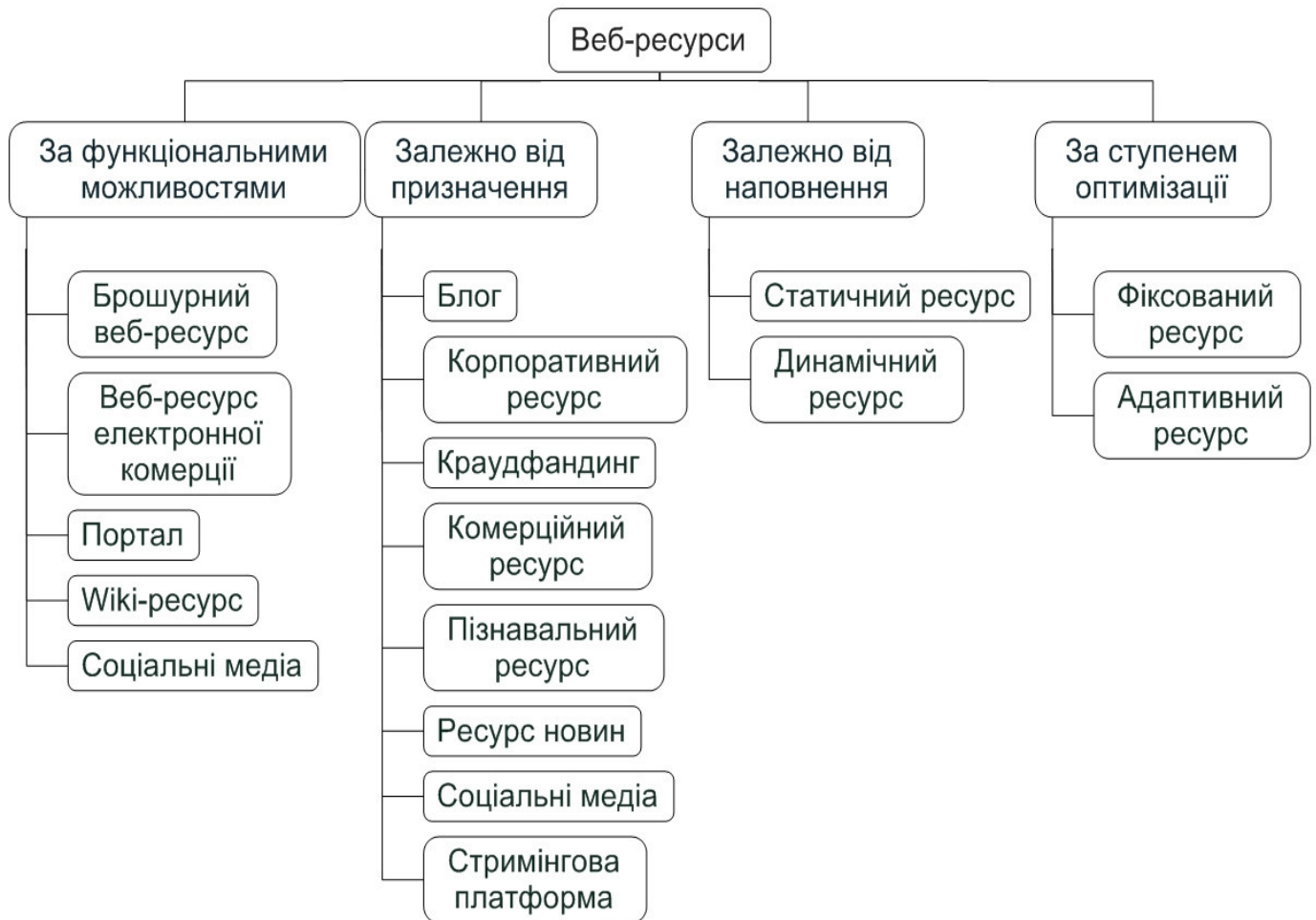


Рис. 1.3. Класифікація веб-ресурсів за Д. Барраклау

Портал є різновидом веб-ресурсу, що надає користувачеві різноманітні інтернет-сервіси (електронне листування, форуми, пошукові системи, новини тощо), які працюють в межах цього ресурсу. Портали є доволі об'ємними ресурсами і допускають наявність великої аудиторії, також вони можуть бути внутрішніми для використання в школі, університеті чи компанії.

Wiki-ресурс дозволяє людям співпрацювати в Інтернеті та писати вміст разом, який надає можливість будь-кому змінювати, додавати та оцінювати вміст наповнення.

Соціальні медіа – це платформи, які дозволяють ділитися текстами, зображеннями та ідеями з усім світом в мережі Інтернет (*Facebook, YouTube, Twitter, Instagram*).

В залежності від типу наповнення, яке використовується при створенні веб-ресурсів, можна розрізнити такі види сторінок: статичні та динамічні [30].

Статичні (фіксовані) характеризуються тим, що їх інформаційне наповнення залишається незмінним для всіх користувачів ресурсом та не регулярно оновлюється зважаючи на те, що зміну контенту можуть здійснювати тільки спеціалісти у веб-технологіях. Статичні веб-ресурси створюються за допомогою простого *HTML*-коду і зазвичай слугують для розповсюдження стабільної інформації.

Динамічні веб-ресурси дозволяють відображати різний вміст кожного разу, коли його відвідують. Це здійснюється за допомогою спеціальних програм, що мають назву скрипти, і формують інформацію в залежності від дій користувача на підставі аналізу його попередніх дій.

З точки зору ступеня оптимізації для різноманітних пристроїв можна поділити на наступні види: фіксований та адаптивний [30].

Фіксований, тобто спроектований на певну ширину пікселів, тому на екранах іншого розміру його перегляд доволі не зручний, оскільки доводиться збільшувати або зменшувати масштаб, щоб побачити вміст сторінки.

Адаптивний ресурс повністю оптимізований для різноманітних електронних пристроїв для коректного відображення вмісту на екрані.

За призначенням веб-ресурси поділяються на [30]:

- блог, що є типом індивідуального веб-ресурсу, який присвячений будь-якій темі та регулярно оновлюється;

- корпоративний ресурс виконує задачі, пов'язані з представництвом будь-якої компанії в Інтернеті, зазвичай, має досить розгалужену структуру та потужну функціональність. Він є обов'язковим атрибутом, що сприяє успішному веденню бізнесу;

- краудфандинговий ресурс слугує для залучення невеликих сум фінансування від зацікавлених в певному проекті людей;
- комерційний ресурс існує у вигляді блогу чи корпоративного веб-ресурсу, ключовим завданням якого є продаж товару чи послуги через Інтернет;
- пізнавальний ресурс надає користувачеві максимально повну інформацію за темою, що його цікавить;
- ресурс новин інформує своїх читачів про поточні справи, хоча певна увага приділяється розважальному контенту;
- соціальні медіа слугують місцем для обміну думками, зображеннями, аудіо чи відео файлами;
- стримінгова платформа виконує завдання трансляції потокового відео за певною тематикою.

В своїх дослідженнях Д. Мілецький (*Jason Miletsky*) [55, с. 39-66] виділяє основні типи веб-ресурсів (рис. 1.4): *B2B*, *B2C*, *C2C*, *B2E*, веб-портали, соціальні мережі, інформаційні ресурси та розважальні ресурси.



Рис. 1.4. Класифікація веб-ресурсів за Д. Мілецьким

B2B (business to business) – це ресурс для взаємодії замовника і постачальника, що включає бізнес-функції, необхідні обом сторонам. Такий інструмент дає

можливості: онлайн-спілкування, ведення звітності, відстеження товарів, обміну документами, додавання акцій, бонусів тощо.

B2C (business to consumer) – це ресурс, що слугує юридичним особам для надання послуг або товарів споживачу. Надає можливість клієнтам за допомогою кошика здійснювати покупку і оплату, а також читати та писати відгуки про товари.

C2C (consumer to consumer) – це ресурс, що слугує комерційній взаємодії між споживачами, які не є юридичними особами. Допомагає здійснювати придбання різноманітних товарів та послуг, пошук роботи або нових співробітників, купівлю та оренду житла тощо.

B2E (business to employee) – це ресурс, що реалізує інтернет-ініціативи для спілкування зі своїми працівниками. Він полегшує внутрішню комунікацію та надає необхідну інформацію співробітникам.

Веб-портал – це ресурс, який слугує користувачам для пошуку певної інформації, яка розташована по всій мережі. Зазвичай це виконується із залученням пошукової системи, яка є програмою, що дозволяє знаходити документи на основі ключових слів, які вводяться у текстове поле. Розрізняють такі види як, розподілений та тематичний веб-портали.

Соціальні мережі – це ресурси, що спеціалізуються на взаємодії людей з різних соціальних груп та географічних місць через Інтернет. Серед послуг, що надаються можна виділити: ведення блогів, відгуки, огляди, дошки оголошень, *RSS* тощо.

Інформаційні ресурси включають в себе текстові, ілюстративні, мультимедійні матеріали або повідомлення інших будь-яких видів, що дозволяють розкрити тематичний зміст ресурсу або одного з його розділів.

Розважальні ресурси функціонують для надання розважальних послуг своїм відвідувачам. Основний контент складає інформація розважального характеру, що включає в себе різноманітне мультимедійне наповнення.

Дослідники С. Майер (*Sven Meyer zu Eissen*) та Б. Стейн (*Benno Stein*) [54, с. 6] проаналізували проблему класифікації з погляду жанрового стилю контенту та сформулювали вісім видів веб-ресурсів (рис. 1.5):

– довідкові ресурси (надають допомогу в пошуках певної інформації);

- інформаційні ресурси (документи з об’ємними фрагментами тексту, такі як статті про дослідження, огляди, технічні звіти чи розділи книг);
- комунікативні ресурси (включають форуми та електронне листування);
- магазини (головною метою яких є продаж продукції та надання інформації про товари);
- неprivатні портали (містять інформацію про компанії, навчальні заклади та інші державні установи);
- приватні портали (типові приватні веб-ресурси з особистим контентом);
- колекції посилань (складаються зі списків посилань за певною тематикою на джерела);
- файлообмінники (надають можливість завантажити різноманітну інформацію та програми).

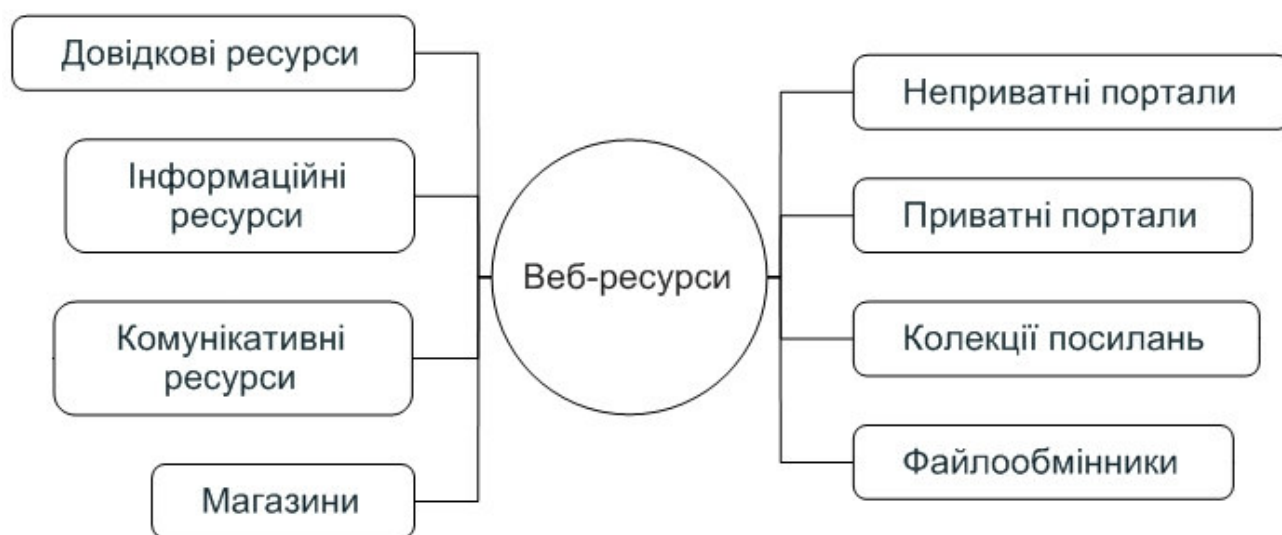


Рис. 1.5. Класифікація веб-ресурсів за С. Майером та Б. Стейном

Отже, розглянувши вище подані класифікації, можна дійти висновку, що кожний дослідник має власне бачення проблеми видо-типологічної класифікації та відповідно виділяє критеріальні показники. Ці показники мають як спільні, так і суттєві відмінності. Слід зазначити, що класифікація веб-ресурсів є необхідною та корисною для всіх користувачів, оскільки надає можливість ефективного пошуку та використання інформації, наявної в Інтернеті. Для виконання дипломної роботи будемо спиратися на видо-типологічну класифікацію Т. Самойлової.

1.2. Принципи проектування веб-ресурсів

В умовах сьогодення кількість веб-ресурсів постійно збільшується з геометричною прогресією, що впливає на якість їхньої побудови та контентного наповнення. У зв'язку з цим проблемою формулювання принципів проектування веб-ресурсів займається велика кількість дослідників, а саме: О. Березко [1], Н. Евдокимов [7], Д. Лещев [11], М. Компанєєтс [8], Д. Бірд (*Jason Beaird*) [31], Ш. Бургшталер (*Sheryl Burgstahler*) [35], Д. Гоув (*Jenny Gove*) [59], К. Дессо (*Christophe Dessaux*) [60], Н. Удай Бхаскар (*N. Uday Bhaskar*) [32], К. Коста (*Carlos J. Costa*) [36], Р. Ноблес (*Robin Nobles*) [22], Ш. Мур (*Shawn Moore*) [56], П. Саху (*Pawan Sahu*) [64], Б. Шнайдерман (*Ben Shneiderman*) [48] та інші.

К. Дессо (*Christophe Dessaux*), Д. Доусон (*David Dawson*), Ф. Філіппі (*Fedora Filippi*) та інші фахівці, що утворили дослідницьку групу під назвою „*Minerva Working Group 5*” сформулювали наступні принципи проектування веб-ресурсів [60, с. 15-63]: прозорість, ефективність, інформаційна підтримка, доступність, орієнтованість на користувачів, реагування, багатомовність, сумісність, захист авторського права, збереженість.

Під принципом прозорості розуміється чітке визначення користувачем назви, характеру та призначення веб-ресурсу, його вміст та ідентифікація керуючої ланки, що відповідальна за його управління [60, с. 15-18].

Основою принципу ефективності є вміст. Веб-ресурс високої якості повинен мати вміст, який належним чином підібраний і відповідно до тематики супроводжується коментарями та підтримуючою інформацією. Також важливим елементом ефективного веб-ресурсу є легкість, з якою користувачі можуть орієнтуватися серед представлених на ньому матеріалах [60, с. 20-24].

Принцип інформаційної підтримки відповідає за надання якості обслуговування користувачів, він орієнтований на контроль якості інформації, що повинна бути актуальною та оновлюваною. Це означає, що вміст веб-ресурсу, який після зазначеної дати стає застарілим або неактуальним, повинен бути частиною архіву [60, с. 27-30].

Принцип доступності фокусується на необхідності обслуговувати всіх членів спільноти користувачів. До них відносяться сліпі та слабоворі користувачі, глухі та слабочуючі, особи з порушеннями рухових нервів, індивідуми з порушеннями читання або з труднощами в навчанні. Існує певний клас технологій, які виступають в якості інтерфейсу між Інтернетом та тими, хто має труднощі при використанні стандартної парадигми доступу клавіатури та миші. Вони включають пристрої користувацького інтерфейсу для заміни тематичної та клавіатурної форми, екрани та клавіатури Брайля, а також програми програмного забезпечення *Wellas*, які „читають” веб-сторінки [60, с. 32-35].

Принцип орієнтованості на користувачів зосереджується на необхідності задоволення перш за все вимог кінцевого споживача. До основних критеріїв цього принципу відносяться: відповідність вмісту ресурсу, зручний користувацький інтерфейс, комфортна навігація, наявність інтерактивних елементів [60, с. 37-40].

Реагування пов'язане з можливістю ресурсу та його власників відповідати на запитання та пропозиції користувачів. Реалізація даного принципу відбувається за допомогою відкритого форуму, на якому будь-які користувачі мають можливість залучатися до обговорення та покращення якості ресурсу. Ресурс, що оперативно реагує на дії відвідувачів, додає цінності та інтересу для кінцевих користувачів через його інтерактивну природу [60, с. 42-44].

Головною метою використання принципу багатомовності є надання інформації максимальній кількості користувачів, адже, чим більша аудиторія, яку охоплює та обслуговує веб-ресурс, тим більш ефективно він виконує свої завдання.

Принцип сумісності розглядає, як веб-ресурс може взаємодіяти з іншими ресурсами. Основна увага приділяється стандартам, які включають в себе методики, технології, моделі даних та інтерфейси, що полегшують взаємодію між веб-ресурсами. Цей принцип передбачає також наявність чітких, автоматизованих пошукових систем, що реалізують віддалений пошук [60, с. 50-53].

Основна проблема принципу захисту авторського права полягає у забезпеченні управління правами інтелектуальної власності та конфіденційності. Застосовується в сферах: захисту прав власника ресурсу від неадекватного

поводження з кінцевим користувачем, захист власника ресурса від судових переслідувань з боку контентного власника, захист конфіденційності кінцевого користувача [60, с. 55-58].

Принцип збереження забезпечує довготривале зберігання інформації, оскільки неможливо з надійністю передбачити технології та підходи, які будуть використовуватися для доступу до інформації в майбутньому [60, с. 60-63].

Н. Удай Бхаскар (*N. Uday Bhaskar*), П. Пратхап Найду (*P. Prathap Naidu*), С. Р. Раві Чандра Бабу (*S. R. Ravi Chandra Babu*) та П. Говіндараджулу (*P. Govindarajulu*) виокремлюють наступні принципи [32, с. 46-59]: естетичність, переналаштованість, контрольованість, корекційність, акцентованість.

Естетичний принцип ґрунтується на візуальному зверненні, що забезпечується дотриманням принципів презентації та графічного дизайну, включаючи забезпечення значущого контрасту між елементами сторінки, створення просторових групувань, вирівнювання елементів сторінок та ефективне використання кольорів. Застосування цього принципу надає веб-ресурсу потужність, функціональність та простоту.

Легка персоналізація та налаштування за допомогою конфігурації лежить в основі принципу переналаштування, оскільки це посилює почуття контролю, призводить до більшої задоволеності користувачів, а також забезпечує особисті вподобання та потреби.

Контрольованість досягається наявністю функціонального інтерфейсу, що викликає відчуття керованості власними діями на веб-ресурсі. Контроль досягається тоді, коли користувач, працюючи у своєму власному темпі, здатен визначити, що робити, як це зробити, а потім легко може це реалізувати.

Користувачі завжди намагаються зрозуміти функціональні можливості ресурсів, при цьому іноді виконують некоректні дії. Принцип корекційності полягає в наявності системи, яка вказує на помилки користувачу та допомагає в їх виправленні шляхом показу підказок, що з'являються на сторінці.

Використання принципу акцентування рекомендує застосовувати техніку візуального наголосу, щоб виділити найважливіші чи найвизначніші частини

сторінки. Виділений елемент повинен бути контрастним, акцентуючи увагу користувача. Для реалізації застосовуються різні методи, такі як: більш висока яскравість, шрифт іншого розміру або накреслення, використання мультимедійних ефектів, контрастний колір, креативне позиціонування.

Існує набір принципів, які можна назвати естетичними принципами, які включають такі критерії, як рівновага, гармонія, лаконічність, передбачуваність та зрозумілість (рис. 1.6), що розглядали такі дослідники як К. Коста (*Carlos J. Costa*), П. Коста (*Pedro Costa*) та М. Апарісіо (*Manuela Aparicio*) [36, с. 2-3].

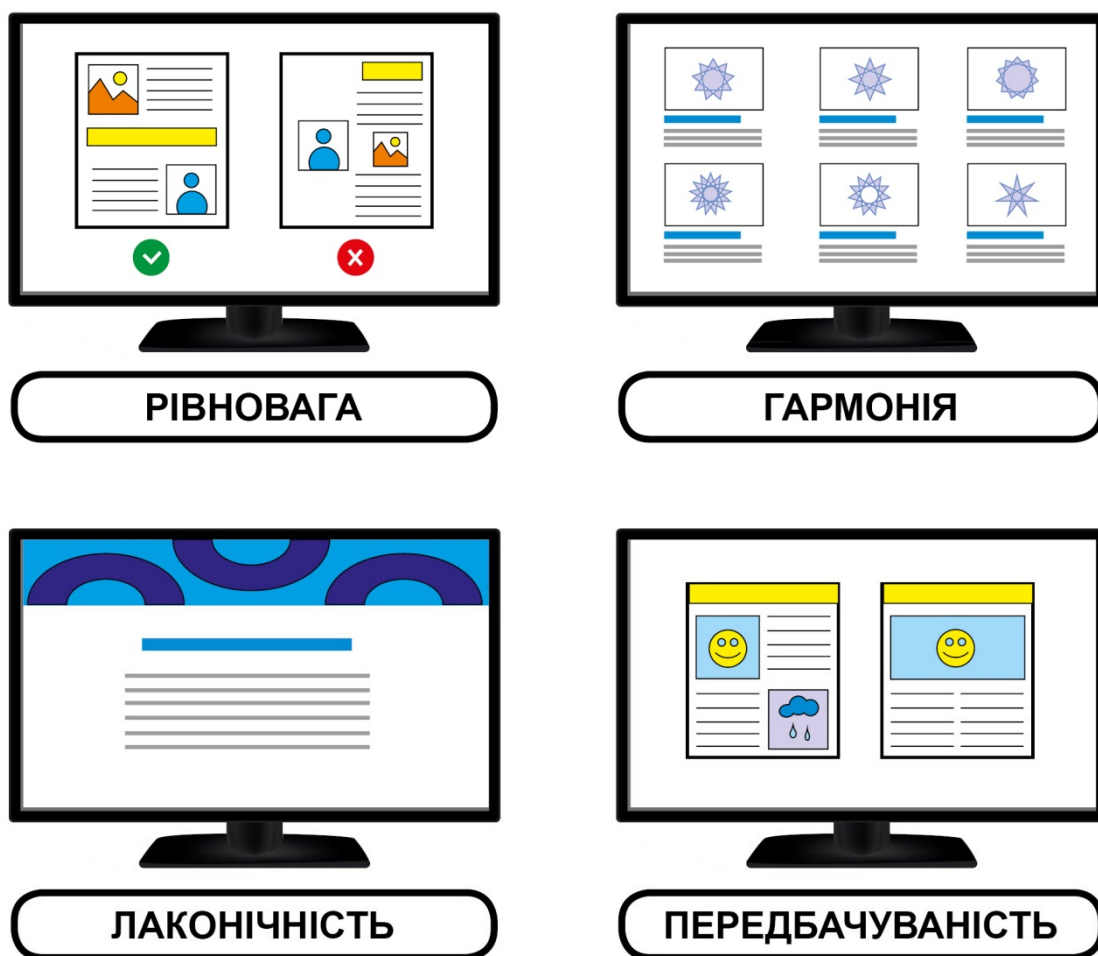


Рис. 1.6. Естетичні принципи

Під принципом рівноваги розуміється правильне розташування різноманітних елементів та їх розміри на сторінках веб-ресурсів, де вони розташовані поруч один біля одного, що утворює єдине ціле, поєднуючись між собою (будь-яка складова не звертає увагу настільки, щоб домінувати над іншими). Якщо сторінка не

врівноважена, вона створює відчуття напруги, що значно погіршує зорове сприйняття.

Гармонійний принцип полягає в створенні єдності за допомогою застосування послідовності та повторюваності в контенті веб-ресурсу, що включає використання подібних шрифтів та кольорів, а також ілюстративний матеріал візуально подібний за тоном та стилем. Візуальна ідентичність є дуже важливою в уніфікованому дизайні ресурсу, оскільки схожість сторінок пов'язує ресурс і створює відчуття єдності.

Хоча привернення уваги є важливою складовою будь-якого веб-ресурсу, але важливо намагатися подати інформацію максимально доступно та зрозуміло. Цьому сприяє використання принципу лаконічності, який полягає в використанні лише необхідної кількості текстового та ілюстративного матеріалу. Зайва графіка може заважати сприйняттю, а надмірна кількість шрифтів або кольорів може відволікати.

Принцип передбачуваності стосується розміщення контенту на кожній сторінці веб-ресурсу в логічно очікуваних місцях. Невідповідності змушують людей витрачати час на спроби зрозуміти як орієнтуватися або яким чином шукати потрібну інформацію. Кожна сторінка веб-ресурсу може відрізнятися за структурою та конструкцією, але ключові елементи повинні бути розташовані на одному місці, щоб користувачі відчували себе комфортно.

Для того, щоб зображення та текст було легко ідентифікувати користувачу застосовується принцип зрозумілості, який полягає в опрацюванні інформації та зведення її до тривіального викладу, уникаючи наукової або сленгової мови.

В теперішній час все більше користувачів використовують для доступу до веб-ресурсів різноманітні мобільні пристрої, такі як телефони та планшети, тому при розробці ресурсів важливо враховувати даний факт. У зв'язку з цим Д. Гоув (*Jenny Gove*) сформулювала основні принципи створення веб-ресурсів для мобільних пристроїв, що були розподілені на три категорії: головна сторінка та засоби навігації веб-ресурсом, пошук по веб-ресурсу, заповнення форм [59].

Як правило, на головній сторінці звичайної версії ресурсу розміщується інформація про власника, область обміну повідомленнями та рекламні матеріали.

Оскільки головна сторінка на мобільному девайсі займає менше простору, то є потреба у відповідних місцях розташувати блоки для того, щоб користувач мав змогу комфортно взаємодіяти з веб-ресурсом (рис. 1.7). Для реалізації даної категорії застосовуються наступні принципи:

- впорядковане та зручне меню (оптимальним варіантом є меню з мінімальною кількістю пунктів, які включають підпункти, що можна відкрити за необхідністю);
- адаптивність сторінок (користувачу на екрані будь-якого розміру не потрібно змінювати масштаб для перегляду інформації);
- зручне повернення на головну сторінку в один дотик.

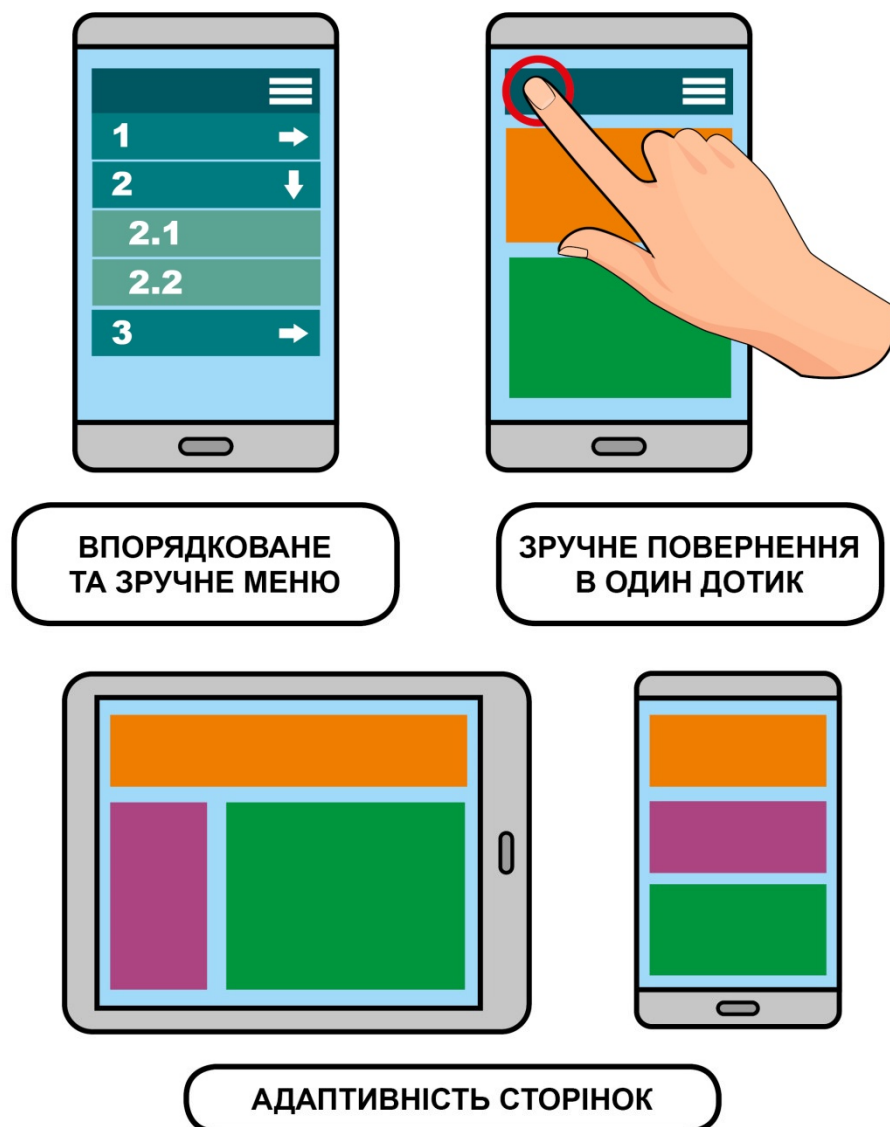


Рис. 1.7. Принципи для головної сторінки та засобів навігації

Ефективний і функціональний пошук має вирішальне значення для допомоги користувачам веб-ресурсів знайти необхідну інформацію (рис. 1.8). Для цього використовують наступні принципи:

- інструмент пошуку необхідно розміщувати на помітному місці;
- наявність функціональних фільтрів для оптимізації пошукових запитів.

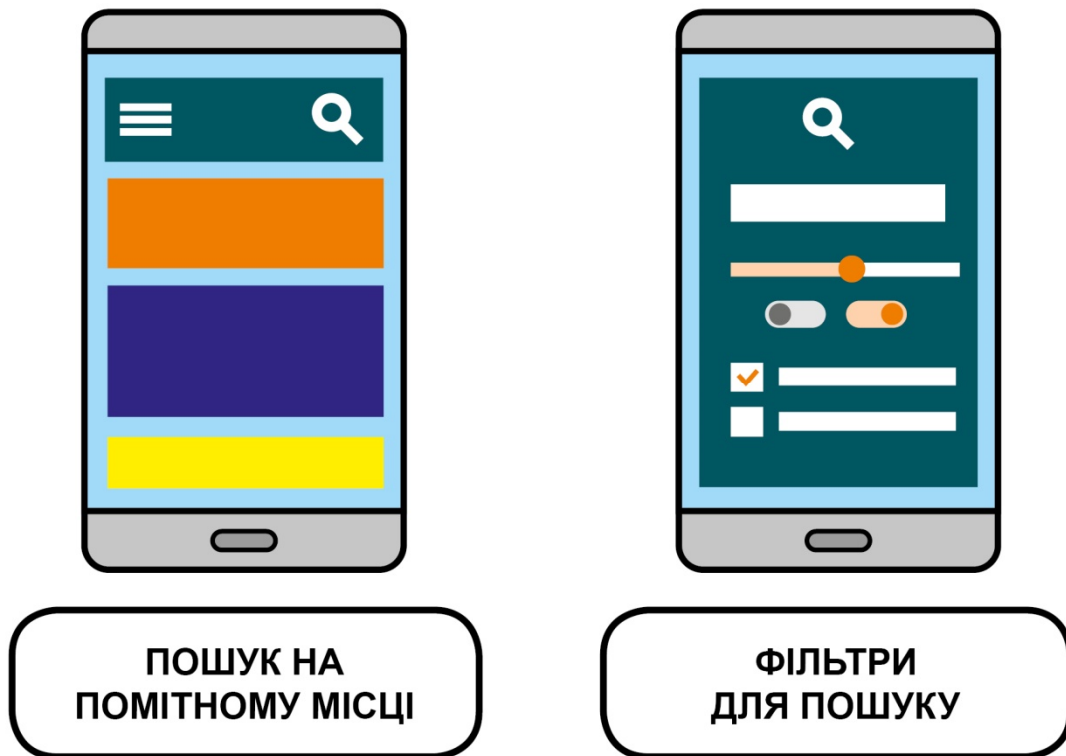


Рис. 1.8. Принципи побудови пошуку

Зручне введення інформації при заповненні форм має велике значення для мобільних пристроїв, оскільки підлаштовує екранну клавіатуру під різні режими введення та проводить базову валідацію даних. Виконанню цих завдань сприяє дотримання таких принципів:

- оптимізація вводу інформації (виклик оптимальної електронної клавіатури);
- ергономічність заповнення форм (під час заповнення поява спливаючих варіантів даних у зручному для користувача вигляді).

Оптимізація пошукових систем (*SEO*) стосується методів, які допомагають веб-ресурсу стати більш помітним для користувачів, які шукають необхідну інформацію за допомогою пошукових систем. Ш. Мур [56] проаналізував дану

проблему та сформував принципи побудови веб-ресурсів з врахуванням *SEO*-стратегії, які стосуються: побудови посилань, прив'язки тексту, щільності ключових слів, альтернативного тексту, оптимальної *URL*-адреси, тегів заголовків та мета-описів.

Принцип побудови посилань завжди є найскладнішою частиною *SEO*, для цього використовуються два види посилань: зворотні та природні [56].

Зворотні посилання – це посилання з інших веб-ресурс на створений веб-ресурс. Кількість зворотних посилань, релевантність сторінок, що посилаються, та ключові слова, які використовуються у зворотних посиланнях (текст прив'язки), впливають на позицію ресурсу в результатах пошуку. Посилання з одного домену враховуються лише один раз.

Природні посилання – це посилання, які надаються веб-ресурсами, що вирішили посилатися на створений ресурс. Розміщуються користувачами або власниками ресурсів, коли вони вважають ресурс або сторінку цікавими і корисними.

Використання прив'язки тексту ставить на меті покращити *SEO*-показники. Текстом прив'язки вважають текст до якого застосовано гіперпосилання, що надає можливість переходити в інші місця веб-ресурсу. Текст прив'язки повинен відповідати сторінці, на яку автор посилається, оскільки слова, що наявні в тексті прив'язки, допомагають підняти рейтинг, який отримає сторінка [56].

Щільність ключових слів – це відсоток випадків, коли ключове слово чи фраза з'являється на веб-сторінці порівняно із загальною кількістю слів на сторінці. У контексті оптимізації пошукової системи щільність ключових слів може бути використана для визначення того, чи відповідає веб-сторінка певному ключовому слову чи фразі ключових слів.

Застосування альтернативного тексту за допомогою тегу *alt* є найважливішим фактором оптимізації зображення. Текст у тегу *alt* відображається замість зображення, якщо зображення не вдається відобразити або завантажується занадто довго.

Оптимально складена *URL*-адреса дає пошуковим системам просте розуміння з вказівками на те, що буде вміщувати веб-сторінка. Хоча використання *URL*-адреси, що використовує ключові слова, може покращити ефективність пошуку на веб-ресурсі, самі *URL*-адреси, як правило, істотно не впливають на здатність сторінки підвищувати рейтинг. Існують наступні основні правила оптимізації *URL*-адреси [56]:

- мають бути максимально простими, релевантними та влучними;
- повинні бути стислими і не перевищувати 2048 символів;
- для розділення слів слід користуватися тільки дефісами;
- для написання адреси слід використовувати тільки прописні літери, щоб уникнути проблеми з повторюваними сторінками.

Теги заголовків мають важливе значення, оскільки титульний елемент сторінки призначений для точного та стислого опису вмісту сторінки. Це важливо як для роботи користувачів, так і для оптимізації пошукових систем. Завжди слід важливі ключові слова робити максимально помітними та стежити за тим, щоб теги заголовків були описовими та читабельними.

Тег мета-опису існує для стислого опису вмісту сторінки. Пошукові системи не задіюють ключові слова або фрази в цьому тезі для ранжування, але мета-описи є основним джерелом для фрагменту тексту, що відображається під списком в результатах. Мета-описи можуть вміщувати в себе будь-яку кількість знаків, проте пошукові системи, зазвичай, вирізають фрагменти довжиною не більше 160 символів, тому бажано залишатися в цьому діапазоні.

Е. Енж (*Eric Enge*) [28, с. 375] сформулював ще один важливий принцип *SEO*-оптимізації під час проектування веб-ресурсів, який носить назву „чотири кліки”. Він полягає в тому, що пошук інформації від головної сторінки ресурсу не має перевищувати чотири кліки мишею. Це допомагає не тільки користувачам, а й пошуковим системам в знаходженні інформації.

Аналіз принципів проектування веб-ресурсів показав, що для виконання завдання дипломної роботи слід використовувати три взаємопов’язані складові проектування: дизайн, програмування та *SEO*-оптимізація.

1.3. Порівняльний аналіз веб-сторінок структурних навчальних підрозділів закладів вищої освіти

Окремі аспекти проблеми дослідження та аналізу веб-сторінок структурних навчальних підрозділів закладів вищої освіти висвітлено в працях І. Мальцева [15], Т. Самойлової [23, 24], Д. Шевченка [27], Г. Браема (*Harald Braem*) [3], Д. Бута (*Danielle Booth*) [34], С. Круга (*Steve Krug*) [10], Е. Мейера (*Eric Meyer*) [17], Ш. Мура (*Shawn Moore*) [56], Р. Томала (*Rafal Tomal*) [25], Д. Чжена (*Jack Zheng*) [73], Б. Шнайдермана (*Ben Shneiderman*) [48] та інших.

Дослідження та аналіз веб-сторінок структурних навчальних підрозділів закладів вищої освіти здійснювалися за наступними критеріями: класифікація ресурсу, структура, кольорова гама, типографіка, архітектура сторінок, *SEO*-оптимізація, адаптивність та кросплатформеність.

Проаналізуємо веб-ресурс кафедри інформатики Оксфордського університету (рис. 1.9), який є структурним навчальним підрозділом даного закладу. Даний ресурс можна віднести: за метою створення до некомерційних; за ступенем доступу до відкритих; за функціональністю до динамічних; за фізичним розташуванням до глобальних; за функціональною спрямованістю до інформаційних [24]. Основною метою ресурса є надання інформації про наукову та навчальну діяльність, актуальні новини, умови вступу та навчання, інноваційні розробки студентів, контактні дані.

Досліджуючи структуру веб-ресурсу, можна виділити наступні компоненти: головна, вступ, дослідницька робота, новини та події, про кафедру, випускники, інновації. Було виявлено, що вона має деревоподібний тип. Відзначимо зручність навігації, яка є відображенням структури та дозволяє швидко знаходити необхідний контент. Суттєвим недоліком є неможливість при наведенні курсором миші переглянути вкладеність рівнів структури [15, с. 60-61].

В дизайні задіяні три основні кольори: темно-синій, світло-коричневий та білий. Темно-синій колір є ключовим в стильовому оформленні, який відображає стабільність та надійність, що є чудовим рішенням для веб-ресурсів навчальних

закладів. Його доповнюють два нейтральні кольори, які створюють закінчену гармонійну композицію [38].

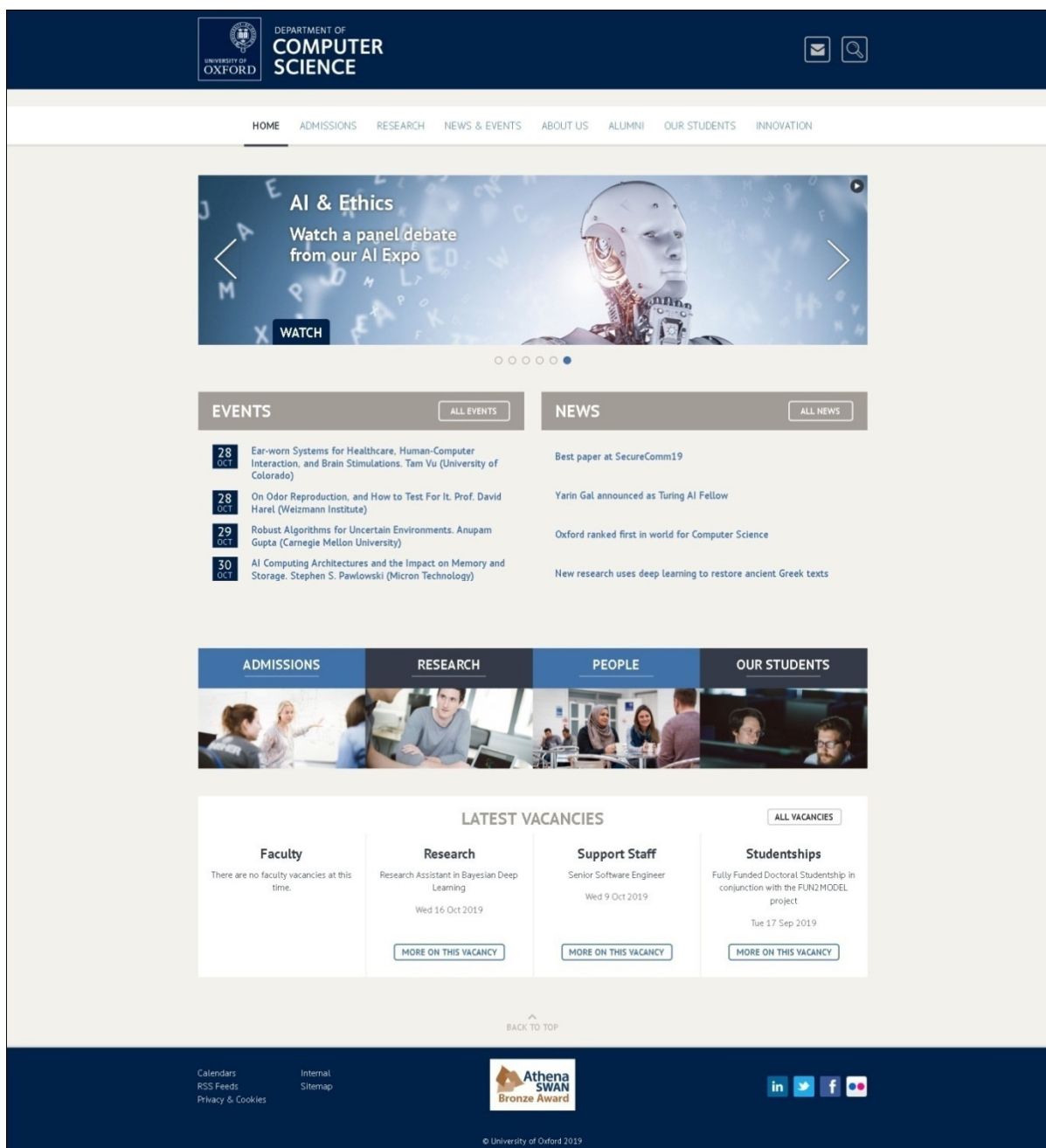


Рис. 1.9. Головна сторінка кафедри інформатики Оксфордського університету

Текстовий контент подано за допомогою шрифту „PT Sans”, який відноситься до шрифтів без засічок сімейства гротесків. Він сприяє легкому сприйняттю інформації без напруження зору [25, с. 69]. Варто зазначити, що розробники передбачили можливість заміни шрифту при його відсутності в програмному забезпеченні на „Helvetica Neue”, „Helvetica”, „Arial”, sans-serif.

В верхній частині хедера розташовано логотип та назву структурного підрозділу, а також присутні дві функціональні інтерактивні піктограми (відповідають за найбільш популярні запити), що розкривають контактну інформацію та вікно пошуку по ресурсу. В нижній частині знаходиться меню, яке вказує на якій сторінці в даний момент знаходиться користувач та можливі варіанти переміщення по веб-ресурсу. Концепційна побудова хедера відповідає принципам ергономічності, оскільки елементи навігації розташовані в постійних та передбачуваних місцях [10, с. 86-98].

Тіло веб-ресурса складається з логічно поділених секцій прямокутної форми, до яких застосовані візуальні ефекти для покращення зорового сприйняття. На головній сторінці знаходиться слайдер, який показує найбільш актуальні та цікаві блоки інформації. Недоліком слайдера можна вважати невдале розташування кнопки переходу. Користувач читає з верхнього лівого кута до нижнього правого, тому найзручніше розміщення функціонального елементу має бути в кінці прочитаного блоку [10, с. 110-117].

Футер розподілений на три блоки: службова інформація (включає календар подій, *RSS*, мапу сайту, конфіденційність та файли *cookie*), копірайт, посилання на соціальні мережі.

Перевіряємо швидкість завантаження веб-ресурсу, оскільки це є важливим критерієм для користувачів з обмеженими можливостями Інтернету [56]. При швидкості Інтернету 100 МБіт/с веб-ресурс завантажився за 2 секунди, що відповідає загальноприйнятим нормам (2-3 секунди). Це свідчить про грамотно написаний код та гарну оптимізацію графічного наповнення.

URL-адреси сторінок відповідають основним правилам оптимізації, тобто вони прості, стислі, розділені дефісами та набрані прописними літерами [56].

Під час перевірки адаптивності виявлено, що веб-ресурс вдало виглядає на екранах різного розміру, зберігаючи свою структуру та функціональні можливості (рис. 1.10). При тестуванні веб-ресурсу на кросплатформенність було виявлено відсутність суттєвих відмінностей у зовнішньому вигляді та функціональних можливостях [17, с. 591-598]. Дослідження адаптивності та кросплатформенності

проводилося за допомогою браузерів *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* та *Opera*.



Рис. 1.10. Адаптивність на мобільному пристрої

Наступним для аналізу обрано веб-ресурс кафедри екології та технології рослинних полімерів Національного технічного університету України „Київський політехнічний інститут імені Ігоря Сікорського” (рис. 1.11). Даний ресурс можна віднести: за метою створення до некомерційних; за ступенем доступу до відкритих; за функціональністю до динамічних; за фізичним розташуванням до глобальних; за функціональною спрямованістю до інформаційних [24]. Основною метою ресурса є надання інформації про наукову та навчальну діяльність, актуальні новини, умови вступу та навчання, підготовчі курси, міжнародне співробітництво та контактні дані.

Досліджуючи структуру веб-ресурсу, можна виділити наступні компоненти: головна, про кафедру, вступ, підготовчі курси, навчання, наукова робота, новини, контакти. Дану структуру відносимо до деревоподібного типу з трьома рівнями вкладеності. Вона сприяє зручній навігації та якісному пошуку інформації завдяки

логічній ієрархічній побудові. Недоліком є графічна подача меню, що відображає структуру веб-ресурсу, через невдало обраний шрифт та малу контрастність [15].

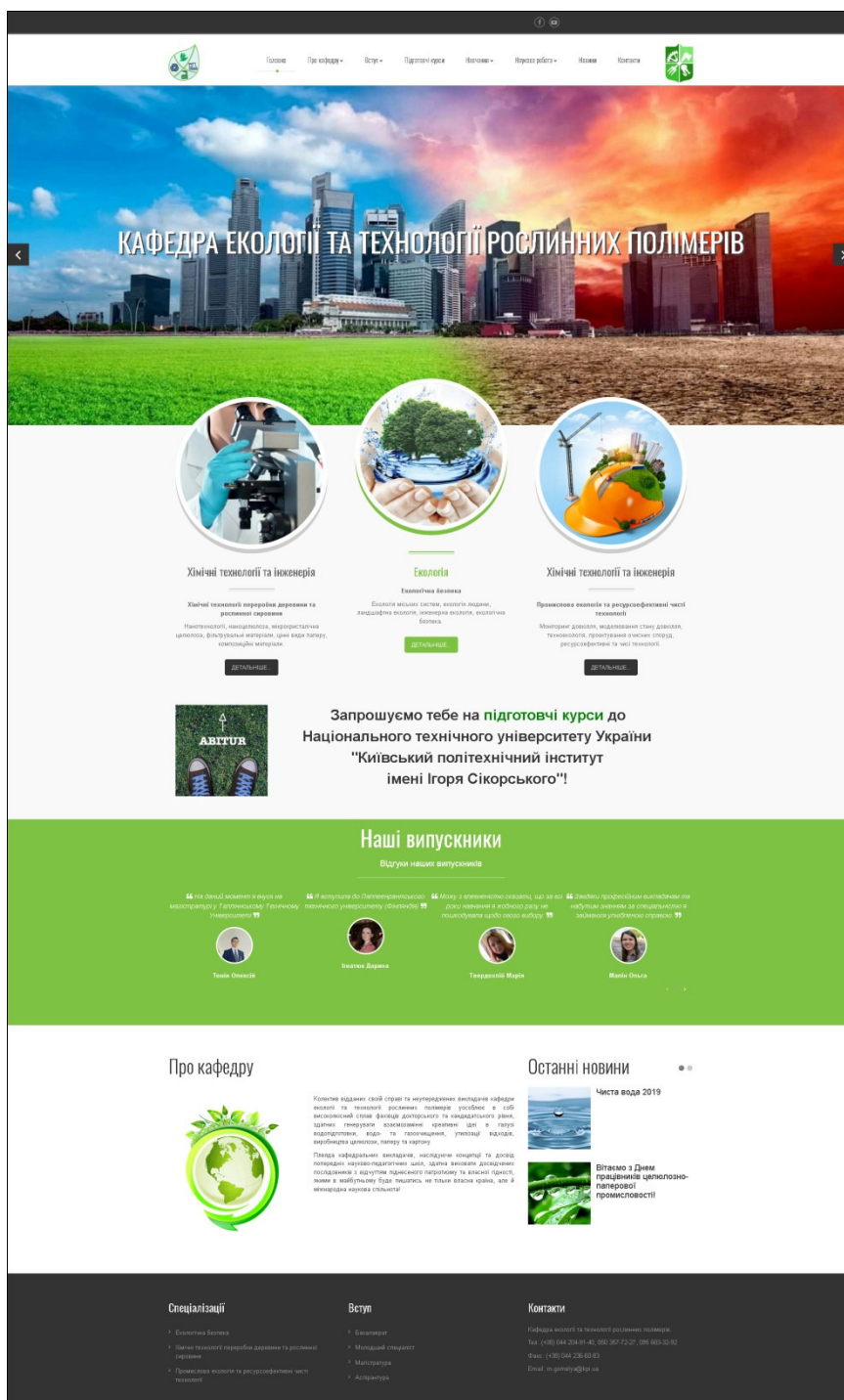


Рис. 1.11. Головна сторінки кафедри екології та технології рослинних полімерів

В дизайні задіяні три основні кольори: зелений, темно-сірий та білий. Кольорову композицію формує білий колір, який є домінуючим та створює відчуття легкості та чистоти. Його гармонійно доповнює зелений колір, який створює

відчуття балансу та стабільності, а також символізує напрямок кафедри на рослинні полімери. Завершуючим штрихом композиції є доповнення елементів в темно-сірому кольорі, що символізує витонченість та професіоналізм [38].

Тексти заголовків подано за допомогою шрифту „*Oswald*”, а для основного тексту застосовано шрифт „*Arial*”. Обидва відносяться до шрифтів без засічок сімейства гротесків. Вони є найбільш поширеними та зручними для сприйняття користувачами [25, с. 69].

Хедер структурно розділений на дві функціональні секції. Верхня секція сірого кольору містить посилання на соціальні мережі у вигляді піктограм. В нижній секції білого кольору розташовано два логотипи між якими розташовано органи навігації, що візуально вказують користувачу на якій сторінці він знаходиться та можливі варіанти переміщення на інші сторінки веб-ресурсу. Хедер сконструйований у відповідності до ергономічних вимог, однак в ньому не дотримані семантичні принципи верстки [10, с. 86-98].

Тіло веб-ресурса складається з логічно поділених секцій, які виділено по чергово двома кольорами, що складає приємне візуальне враження. В дизайнерському оформленні блоків примінено два види геометричних форм (коло та прямокутник) до яких застосовано анімаційні ефекти. На головній сторінці знаходиться слайдер, який виконує тільки декоративну роль, не надаючи можливості користувачу перейти на сторінку з інформацією, анонсовану слайдером [10, с. 110-117].

Футер веб-ресурсу розділений на дві секції, де верхня включає в себе посилання на три інформаційні блоки (спеціалізація, вступ та контакти), а нижня містить дані про захист авторського права.

Швидкість завантаження веб-ресурсу зі швидкістю Інтернету 100 МБіт/с склала 2.5 секунди, що лежить в межах допустимої норми. Отже, робимо висновок про грамотно зверстаний код та оптимізований графічний матеріал [56].

URL-адреси сторінок відповідають основним правилам оптимізації, тобто вони прості, стислі, розділені дефісами та набрані прописними літерами [56].

Перевіряючи веб-ресурс на адаптивність виявлено, що інформаційне наповнення вірно масштабується під різні розміри екрану. Але у виявлено значний недолік у адаптиві пунктів меню через те, що руйнується структура рівнів вкладеності і користувачу важко зорієнтуватися в ньому (рис. 1.12). При тестуванні веб-ресурсу на кросплатформенність було з'ясовано, що в сучасних версіях браузерів *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* та *Opera* відсутні суттєві відмінності у зовнішньому вигляді, але в *Internet Explorer* версій 9.0 та нижче з'являються проблеми з коректним відображенням інформації [17, с. 591-598].

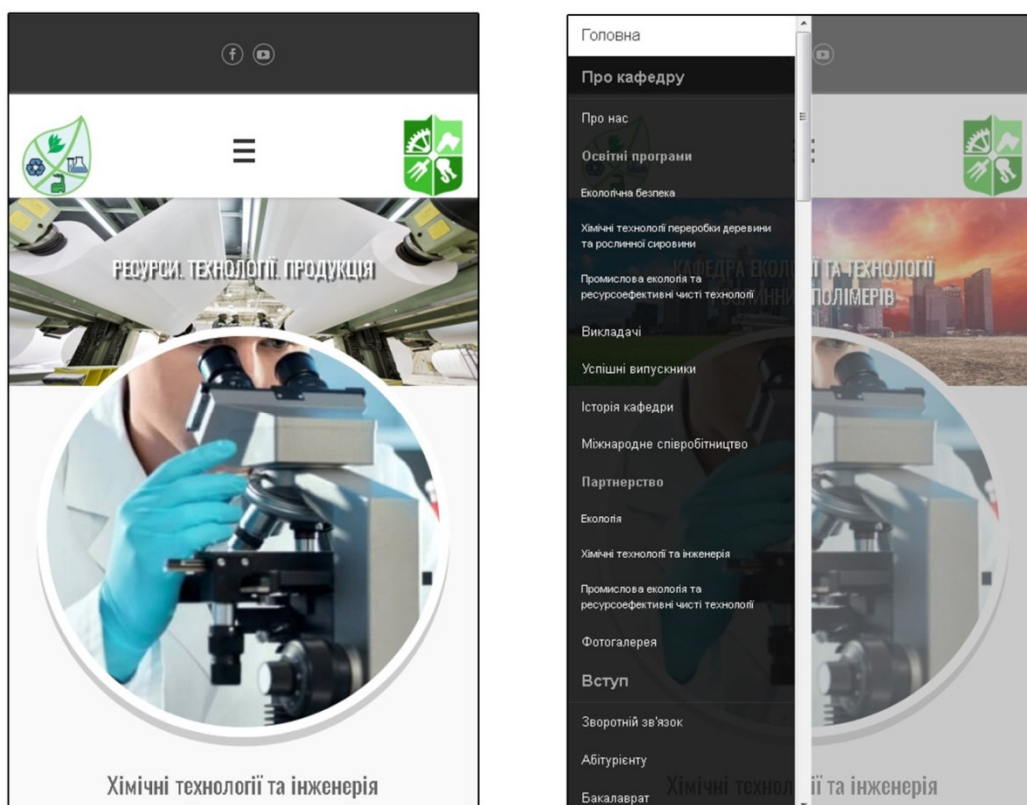


Рис. 1.12. Тестування адаптивності

Останнім для аналізу обрано веб-ресурс кафедри мистецтва та історії мистецтв Стенфордського університету (рис. 1.13). Даний ресурс можна віднести: за метою створення до некомерційних; за ступенем доступу до напіввідкритих; за функціональністю до динамічних; за фізичним розташуванням до глобальних; за функціональною спрямованістю до інформаційних [24]. Основною метою ресурса є надання інформації про наукову та навчальну діяльність, актуальні новини та події, умови вступу та навчання, творчі роботи та виставкові заходи, контактні дані.

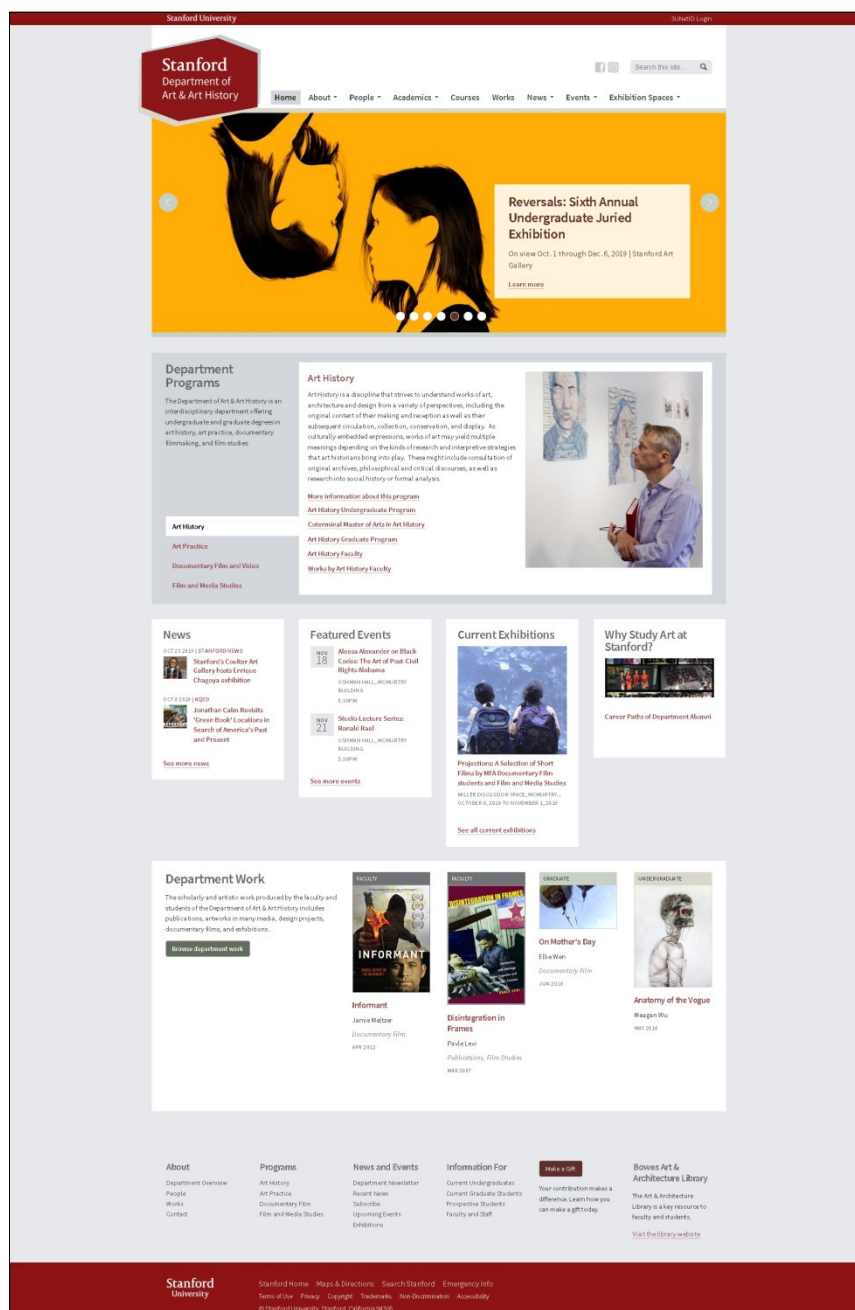


Рис. 1.13. Головна сторінка кафедри мистецтва та історії мистецтва

Досліджуючи структуру веб-ресурсу, можна виділити наступні компоненти: головна, про кафедру, інформація про викладачів та студентів, навчальні програми, курси, творчі напрацювання, новини, події та виставки. Веб-ресурс має деревоподібний тип структури з двома рівнями вкладеності, що є досить зручним варіантом навігації по інформації [15, с. 60-61].

Кольорова концепція дизайну утворена за допомогою комбінації трьох кольорів: темно-червоного, різних відтінків сірого та білого. Темно-червоний, який в

невеликій кількості слугує акцентуванню на ключових елементах, в поєднанні з білим і сірим надає сайту елегантності і професіоналізму [38].

Текстовий контент подано за допомогою шрифту „*Source Sans Pro*”, який відноситься до шрифтів без засічок сімейства гротесків. Він забезпечує легке сприйняття текстової інформації користувачем [25, с. 69]. Варто зазначити, що розробники передбачили можливість заміни шрифту при його відсутності в програмному забезпеченні на „*Arial*”, „*Helvetica*”, *sans-serif*.

Хедер веб-ресурсу розділений на три функціональні блоки. Верхній блок надає можливість користувачу перейти на офіційний сайт Стенфордського університету та містить текстове гіперпосилання для переходу на сторінку авторизації. Середній блок включає в себе пошук по ресурсу та посилання у вигляді піктограм на соціальні мережі. Нижній блок відведений навігації по веб-ресурсу, яка відповідає структурі ресурсу. Об'єднуючим елементом середнього та нижнього блоків виступає логотип кафедри. Можна відзначити, що хедер вдало побудований та зручний в користуванні [10, с. 86-98].

Тіло веб-ресурса складається з окремих інформаційних блоків, які відповідають дизайнерській концепції, мають прямокутну форму та виділені різними відтінками сірого та білого кольору [10, с. 110-117]. На головній сторінці знаходиться слайдер, який анонсує найбільш важливі події в яких приймає участь кафедра.

Футер веб-ресурсу розділений на два блоки, де верхня частина частково дублює навігацію, містить посилання на бібліотеку університету та посилання на сторінку благодійних внесків для розвитку кафедри. Нижній блок містить інформацію Стенфордського університету стосовно: розташування корпусів та маршрутів проїзду до них, інструкції для надзвичайних ситуацій, правила безпечного користування ресурсом та захист авторського права.

Тестування швидкості завантаження веб-ресурсу виявило, що воно знаходиться в межах допустимої норми та складає 1.5 секунди при швидкості Інтернету 100 МБіт/с. Така швидкість обумовлюється використанням технологій кешування та оптимізації складових веб-ресурсу [56].

URL-адреси сторінок відповідають основним правилам оптимізації, тобто вони прості, стислі, розділені дефісами та набрані прописними літерами [56].

Під час перевірки адаптивності виявлено, що веб-ресурс вдало виглядає на екранах різного розміру, зберігаючи свою структуру та функціональні можливості (рис. 1.14). Слід відзначити вдало структуроване меню, що відповідає сучасним вимогам до адаптації. При тестуванні веб-ресурсу на кросплатформенність було виявлено відсутність суттєвих відмінностей у зовнішньому вигляді та функціональних можливостях [17, с. 591-598]. Дослідження адаптивності та кросплатформенності проводилося за допомогою браузерів *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* та *Opera*.

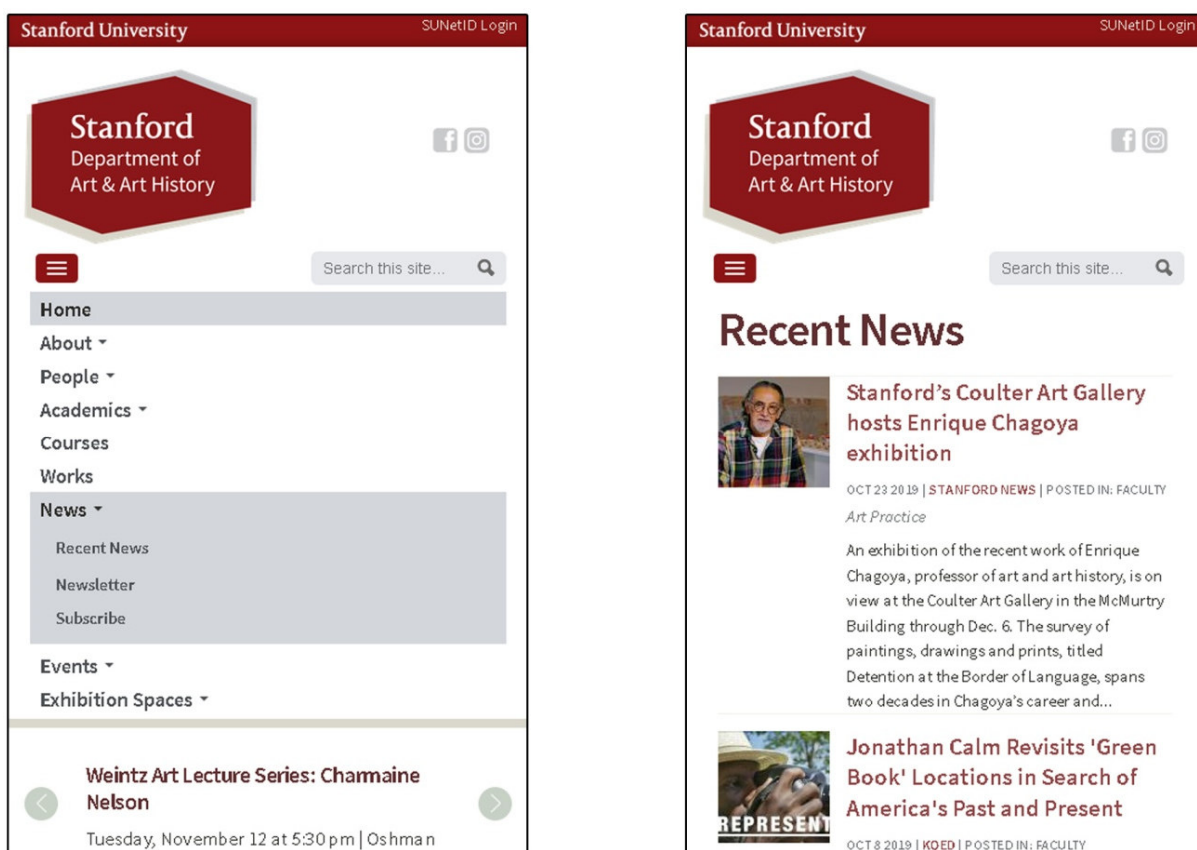


Рис. 1.14. Тестування адаптивності

Для порівняльного аналізу було обрано веб-сторінки структурних навчальних підрозділів трьох провідних закладів вищої освіти. Дослідження проведено за обраними критеріями та виявлено, що всі вони в основному відповідають вимогам проектування, але мають незначні структурні помилки.

Висновки до розділу

На основі аналізу наукових праць дослідників, що вивчали проблему видотипологічної класифікації веб-ресурсів, можна дійти висновку про відсутність єдиного підходу. В дипломній роботі у якості робочої класифікації використовуємо класифікацію Т. Самойлової. За видовими ознаками веб-ресурси розподіляються на: комерційні, корпоративні та інформаційні ресурси. За типологічними ознаками веб-ресурси розподіляються: залежно від мети створення, за ступенем доступу, за функціональністю та стилем оформлення, за фізичним розташуванням та за функціональною спрямованістю.

Принципи проектування веб-ресурсів базуються на трьох взаємопов'язаних складових: дизайні, програмуванні та *SEO*-оптимізації. Дизайн ґрунтується на візуальному зверненні, що забезпечується дотриманням принципів: прозорості, ефективності, орієнтованості на користувача, естетичності, акцентування, рівноваги, гармонії, лаконічності та передбачуваності. Програмування забезпечує функціональні можливості за допомогою принципів: інформаційної підтримки, доступності, реагування, багатомовності, сумісності, збереження, персоналізації, контрольованості, корекційності та адаптивності. *SEO*-оптимізація допомагає підвищити рейтинг веб-ресурсу серед пошукових систем за допомогою принципів: побудови посилань, прив'язки тексту, щільності ключових слів, альтернативного тексту, оптимальної *URL*-адреси, тегів заголовків та мета-описів.

Для порівняльного аналізу обрано веб-сторінки структурних навчальних підрозділів провідних закладів вищої освіти: Оксфордського університету, Стенфордського університету та Національного технічного університету України „Київський політехнічний інститут імені Ігоря Сікорського”. Дослідження проводилось за наступними критеріями: класифікація ресурсу, структура, кольорова гама, типографіка, архітектура сторінок, *SEO*-оптимізація, адаптивність та кросплатформеність. Було виявлено, що всі розглянуті веб-сторінки в цілому відповідають вимогам та принципам проектування, але мають незначні структурні помилки.

РОЗДІЛ 2

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ ВЕБ-РЕСУРСУ

2.1. Порівняльний аналіз програмного забезпечення

Для написання оптимізованого та безпомилкового коду розробнику веб-ресурсу необхідно вибрати відповідне програмне забезпечення, яке розділяється на дві категорії: текстові редактори та інтегровані середовища розробки. Текстовий редактор – це комп’ютерна програма, яка призначена для створення та редагування текстових даних. До популярних текстових редакторів відносяться: *Sublime Text 3*, *Visual Studio Code*, *Atom*, *Brackets*, *Notepad++*, *Vim*, *TextMate*. Інтегроване середовище розробки (*IDE*) – це комплексна комп’ютерна програма для розробки програмного забезпечення, що включає в себе наступні складові: текстовий редактор, компілятор чи інтерпретатор, а також інструменти для автоматизації процесів написання та відлагодження програм. До популярних *IDE* відносяться: *Jetbrains Webstorm*, *Adobe Dreamweaver*, *Microsoft Visual Studio*, *NetBeans*, *IntelliJ IDEA* [42]. Вірно підібране програмне забезпечення підвищує ефективність написання та редагування коду, а також запобігає уникненню різноманітних помилок.

При виконанні дипломної роботи для створення веб-ресурсу кафедри доцільніше використовувати текстовий редактор, оскільки на відміну від *IDE*, він працює набагато швидше (відсутня велика кількість специфічних функцій, які завантажують апаратне забезпечення), має логічний та простий інтерфейс, краще зарекомендував себе при роботі з невеликими проектами (10-20 файлів) [42].

Під час вибору текстового редактору необхідно враховувати наступні основні критерії [18]:

- підсвічування синтаксису (виділення початку та закінчення окремого логічного блоку певним кольором сприяє візуальному пошуку та робить код зручним для читання та корегування);

- вимоги до апаратного забезпечення (мінімальні системні вимоги для комфортної та швидкої роботи);
- автоматичні відступи (перенесення рядка програмою з автоматичним виставленням рівня вкладеності для покращення орієнтації в структурі коду);
- автодоповнення (аналіз програмою написаного коду та пропозиція варіантів подальшого продовження);
- розділення на робочі зони (можливість розділення робочої області редактора на необхідну кількість секторів);
- міні-мапа (відображення структури файлу та зручна навігація по документу);
- персоналізація інтерфейсу (можливість індивідуального налаштування під користувача);
- інтеграція модулів та додатків (покращення функціональності завдяки додавання плагінів);
- гарячі клавіші (наявність комбінацій клавіш для швидкого виконання певних дій та можливість налаштування під розробника).

К. Вейзіл (*Catalin Vasile*) [68] досліджуючи сферу популярних текстових редакторів для написання коду, сформував перелік найбільш зручних та популярних програм. Для аналізу та вибору редактору обираємо трійку лідерів із списку: *Visual Studio Code*, *Sublime Text 3* та *Atom*.

Visual Studio Code – це безкоштовний кросплатформений редактор вихідного коду, розроблений компанією *Microsoft*. Інтерфейс програми включає чотири області: робочу, бічну панель, рядок стану та панель перегляду (рис. 2.1). Під час запуску програми папка та робочі файли відкриваються в тому ж стані, в якому вони були при закритті. Редактор оснащений механізмом автозаповнення *IntelliSense*, що надає можливість допису назви оголошених змінних. Базова комплектація включає в себе значну частину функціоналу *IDE*, інтеграцію з *Git*, а також є можливість встановлення допоміжних модулів і додатків. В текстовому редакторі вбудовано відладчик коду при роботі з *JavaScript*, *TypeScript*, *Node.js*. Програма має відкритий вихідний код, що надає можливість фахівцям коректувати редактор під себе [43].

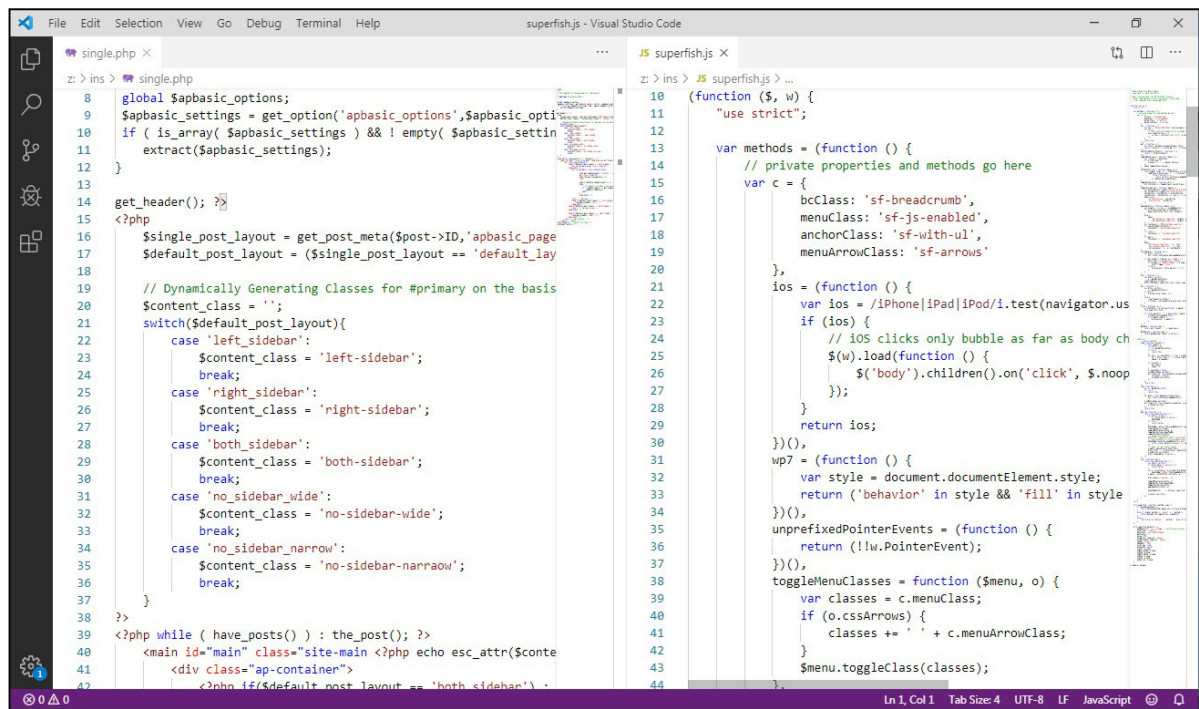


Рис. 2.1. Інтерфейс *Visual Studio Code*

Sublime Text 3 – багатифункціональний текстовий редактор розроблений компанією *Sublime HQ Pty Ltd*. Інтерфейс програми включає чотири області: робочу, бічну панель, рядок стану та панель перегляду (рис. 2.2). Кросплатформеність даного редактору дозволяє працювати з усіма операційними системами на базі *Windows*, *macOS* та *Linux*. Програма не виставляє високі вимоги до апаратного забезпечення, тому дуже швидко завантажується та працює. Існує портативна версія редактора, що не потребує інсталяції, а отже є можливість запуску з будь-якого носія інформації на сторонньому комп'ютері. Також існує велика кількість доповнень з відкритим вихідним кодом, що дозволяє модифіковувати їх під особисті завдання. Для зручності роботи в даному продукті передбачена можливість розділення робочої області на необхідну кількість секцій [40].

Atom – це текстовий редактор з відкритим вихідним кодом розроблений компанією *GitHub*. Він працює на операційних системах *Windows*, *macOS* та *Linux*, а отже є кросплатформеним. Інтерфейс програми включає чотири області: робочу, бічну панель, рядок стану та панель перегляду (рис. 2.3). Текстовий редактор поставляється з 81 вбудованим пакетом, але надається можливість додатково

інсталювати доповнюючий інструментарій. Програма має відкритий вихідний код, що надає можливість фахівцям коректувати редактор під себе. Продукт підтримує плагін *Teletype*, що дозволяє працювати над кодом, взаємодіючи з іншими розробниками в режимі реального часу. Завдяки розумному механізму автозаповнення редактор сприяє швидкому набору коду [29].

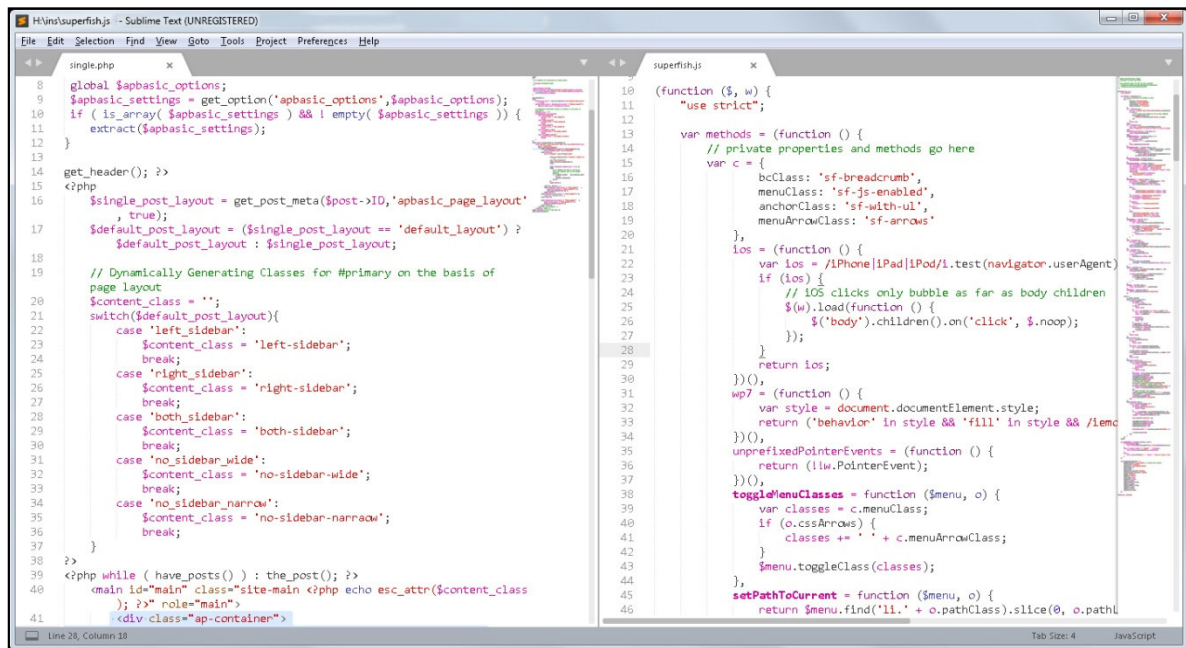


Рис. 2.2. Інтерфейс *Sublime Text 3*

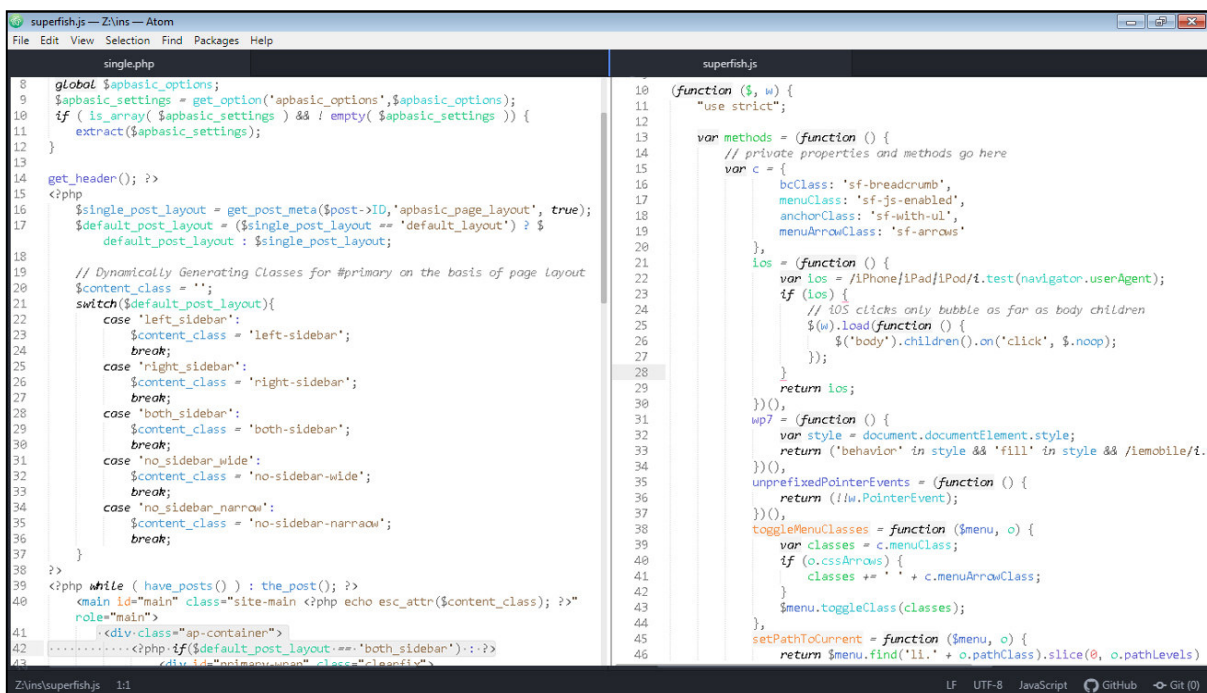


Рис. 2.3. Інтерфейс *Atom*

Для вирішення задачі вибору текстового редактора порівнюємо три програми: *Visual Studio Code*, *Sublime Text 3* та *Atom* (табл. 2.1) [29, 40, 43].

Таблиця 2.1

Порівняльний аналіз текстових редакторів

Параметри		<i>Visual Studio Code</i>	<i>Sublime Text 3</i>	<i>Atom</i>
Вимоги до апаратного забезпечення	частота процесора, <i>GHz</i>	1.6	1.0	1.6
	оперативна пам'ять, <i>MB</i>	1000	512	1000
Кросплатформеність		<i>Windows, macOS, Linux</i>	<i>Windows, macOS, Linux</i>	<i>Windows, macOS, Linux</i>
Портативна версія		Відсутня	Присутня	Відсутня
Підсвічування синтаксису		Присутнє	Присутнє	Присутнє
Автоматичні відступи		Присутні	Присутні	Присутні
Автодоповнення		Присутнє	Частково присутнє	Присутнє
Розділення на робочі зони		Присутнє	Присутнє	Присутнє
Міні-мапа		Присутня	Присутня	Встановлюється за допомогою плагіна
Персоналізація інтерфейсу		Присутня	Присутня	Присутня
Інтеграція модулів та додатків		Присутня	Присутня	Присутня
Відладчик		Присутній	Встановлюється за допомогою плагіна	Присутній
Гарячі клавіші		Присутні	Присутні	Присутні
Пошук та заміна (<i>regex</i>)		Присутні	Присутні	Присутні
Сумісна робота над проектом		Присутня	Встановлюється за допомогою плагіна	Присутня
Сніппети		Присутні	Присутні	Присутні
Ціна ліцензії, \$		Безкоштовна	70 (існує безкоштовна версія з урізаним функціоналом)	Безкоштовна

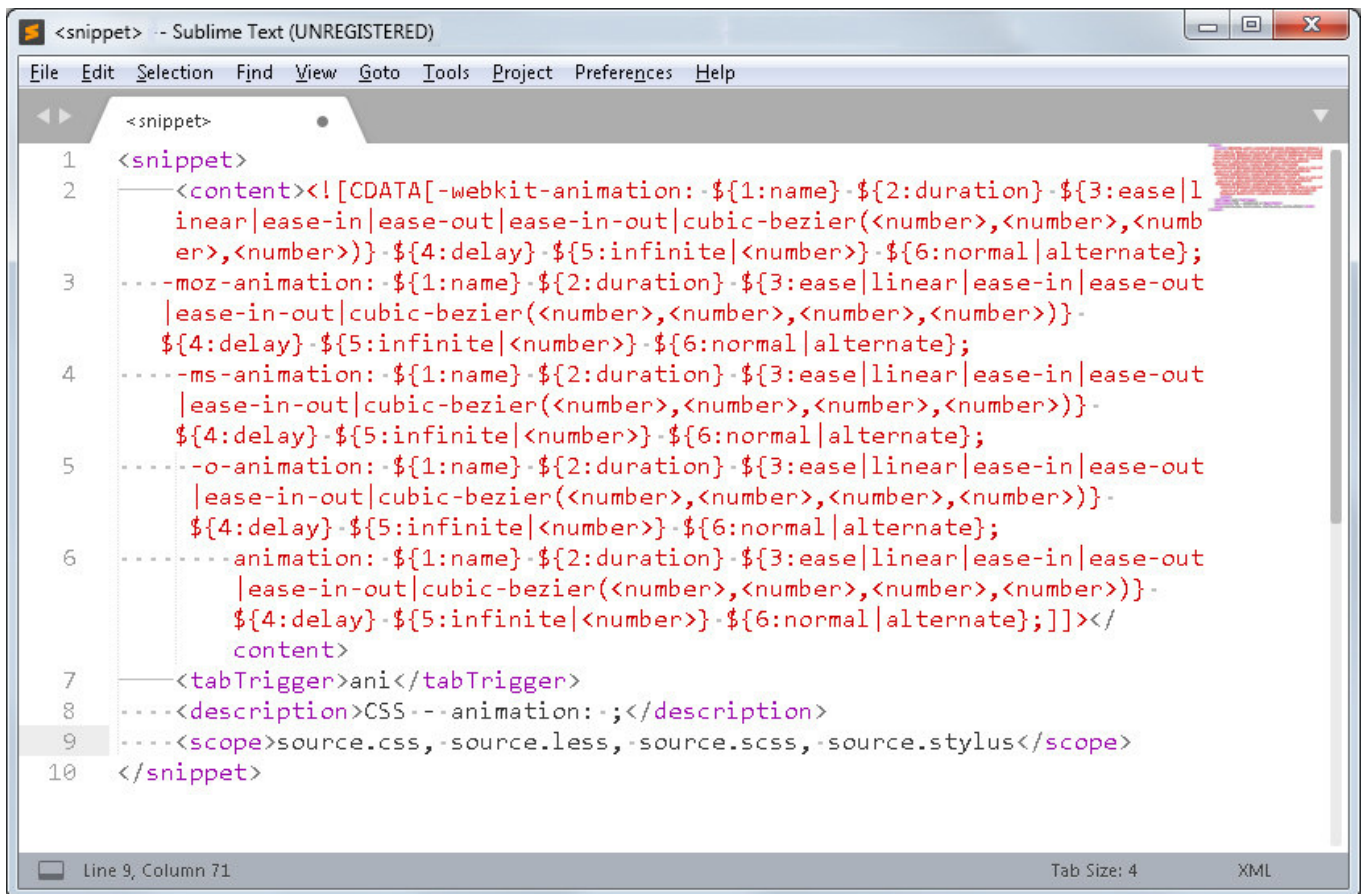
З проведеного огляду та аналізу видно, що всі розглянуті редактори мають майже однакові функціональні можливості, за виключенням незначних інструментів, які додатково можна встановити за допомогою плагінів та доповнень. Для виконання дипломної роботи було обрано текстовий редактор *Sublime Text 3*. На користь вибору вплинули наступні фактори: системні вимоги до апаратного та програмного забезпечення, наявність портативної версії, зручність інтерфейсу та багаторічний досвід у роботі з даним редактором.

Оскільки базова версія *Sublime Text 3* не має всіх необхідних інструментальних компонентів для виконання дипломної роботи, вирішено вдосконалити продукт шляхом інсталяції додаткових можливостей [51]. Було встановлено наступні плагіни:

- *emmet* (дозволяє пришвидшити написання коду шляхом вставки фрагментів коду за допомогою спеціального синтаксису);
- *advanced new file* (покращує створення файлів);
- *side bar enhancements* (додає новий функціонал для сайдбару);
- *colorPicker* (надає можливість обрання кольору за допомогою палітри);
- *sublimeCodeIntel* (спрощує пошук фрагментів раніше написаного коду);
- *sublime linter* (вказує на синтаксичні помилки);
- *color highlighter* (візуальне відображення кольору);
- *liveReload* (зміни коду одразу відображаються в браузері).

Подальша модернізація програми включає написання сніпсетів, які являються готовими блоками коду та часто використовуються при написанні часто вживаних елементів на веб-ресурсі. Створюємо сніпсети з використанням префіксів для забезпечення коректного відображення в усіх браузерах анімацій, переходів та декоративних елементів (рис. 2.4).

Для процесів оптимізації *HTML*, *CSS*, *JavaScript* та ілюстративного матеріалу комбінуємо *Gulp* із *Sublime Text 3*. *Gulp* мінімізує файли з метою зменшення їх розміру, що сприяє більш швидкому завантаженню. Він допомагає тестувати код на помилки вказуючи на фрагменти коду для валідації [49].



```
1 <snippet>
2   <content><![CDATA[-webkit-animation: -${1:name}-${2:duration}-${3:ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier(<number>,<number>,<number>,<number>) }-${4:delay}-${5:infinite|<number>}-${6:normal|alternate};
3   ---moz-animation: -${1:name}-${2:duration}-${3:ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier(<number>,<number>,<number>,<number>) }-${4:delay}-${5:infinite|<number>}-${6:normal|alternate};
4   ----ms-animation: -${1:name}-${2:duration}-${3:ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier(<number>,<number>,<number>,<number>) }-${4:delay}-${5:infinite|<number>}-${6:normal|alternate};
5   -----o-animation: -${1:name}-${2:duration}-${3:ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier(<number>,<number>,<number>,<number>) }-${4:delay}-${5:infinite|<number>}-${6:normal|alternate};
6   -----animation: -${1:name}-${2:duration}-${3:ease|linear|ease-in|ease-out|ease-in-out|cubic-bezier(<number>,<number>,<number>,<number>) }-${4:delay}-${5:infinite|<number>}-${6:normal|alternate};]]></content>
7   <tabTrigger>ani</tabTrigger>
8   <description>CSS--animation: -;</description>
9   <scope>source.css,-source.less,-source.scss,-source.stylus</scope>
10 </snippet>
```

Рис. 2.4. Кросплатформений сніппет для анімації

Порівняльний аналіз програмного забезпечення для написання коду проводився за наступними ключовими критеріями: вимоги до апаратного забезпечення, підсвічування синтаксису, автоматичні відступи, автодоповнення, розділення на робочі зони, наявність міні-мапи, персоналізація інтерфейсу, інтеграція модулів та додатків, гарячі клавіші. На основі цього було обрано текстовий редактор *Sublime Text 3* із подальшим вдосконаленням його функціональних можливостей за допомогою плагінів та написання сніппетів.

Для опрацювання графічного матеріалу та створення дизайнерської концепції розробнику веб-ресурсу слід обрати відповідне програмне забезпечення, яке володіє повним пакетом функціональних можливостей. Враховуючи поставлені задачі, доцільніше використовувати векторний редактор, оскільки складові елементи оформлення веб-ресурсу у векторній формі не будуть втрачати якість при будь-яких змінах розмірів зображення.

Під час вибору векторного графічного редактору необхідно враховувати наступні основні критерії [33]:

- функціональність інструментів (можливість налаштування інструментів під різні типи задач);
- підтримка кольорових моделей (наявність всіх популярних моделей);
- швидкість роботи;
- можливість роботи з популярними форматами;
- зручність інтерфейсу (ергономічність панелі та здатність до персоналізації);
- налаштування експорту файлів (присутність великої кількості налаштувань при збереженні);
- можливості роботи з текстом (здатність налаштовувати параметри текстових даних).

Т. Болт (*Thomas Boldt*) [33] аналізуючи популярні векторні графічні редактори, склав перелік найбільш зручних програм. Для аналізу та вибору редактору обираємо лідерів із списку: *Adobe Illustrator* та *CorelDRAW*.

Adobe Illustrator – професійний графічний редактор для створення та макетування ілюстративного матеріалу, розроблений компанією *Adobe*. Інтерфейс програми включає велику кількість інструментарію та налаштувань (рис. 2.5). Програма дозволяє створювати *actions* для швидкого застосування необхідних маніпуляцій над об'єктом роботи. Графічний редактор оснащений зручною інтуїтивно-зрозумілою панеллю шарів, що спрощує роботу з фрагментами файлу та сприяє орієнтації по документу [69].

CorelDRAW – це пакет професійних інструментів для редагування графічного матеріалу, розробки дизайну і створення макетів, розроблений компанією *Corel*. Інтерфейс програми виконаний в мінімалістичному стилі з можливістю його персоналізації під конкретні завдання (рис. 2.6). Програма має велику варіативність налаштувань інструментів та дозволяє повноцінно працювати з текстовими масивами інформації. *CorelDRAW* володіє потужними та зручними засобами трансформації для зміни форми об'єкту. В пакеті присутня можливість редагування фотографій за допомогою вбудованих фільтрів [70].

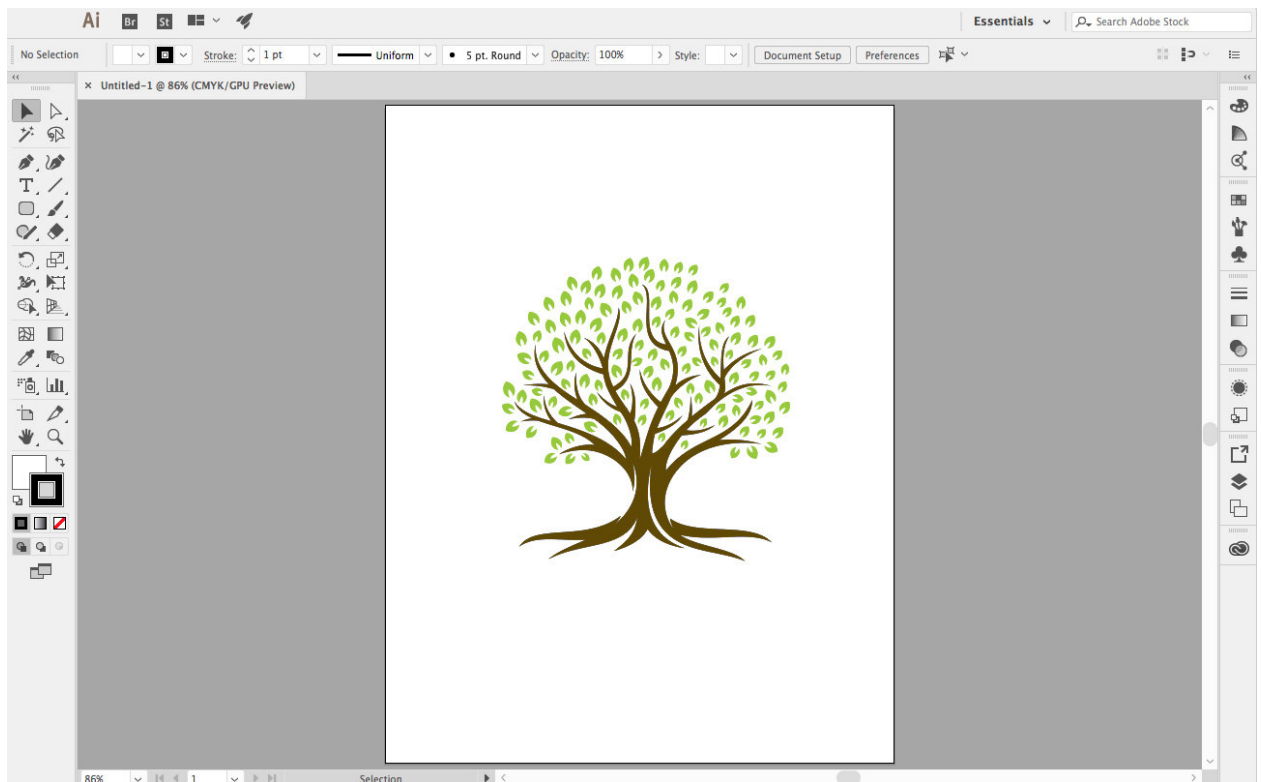


Рис. 2.5. Інтерфейс *Adobe Illustrator*

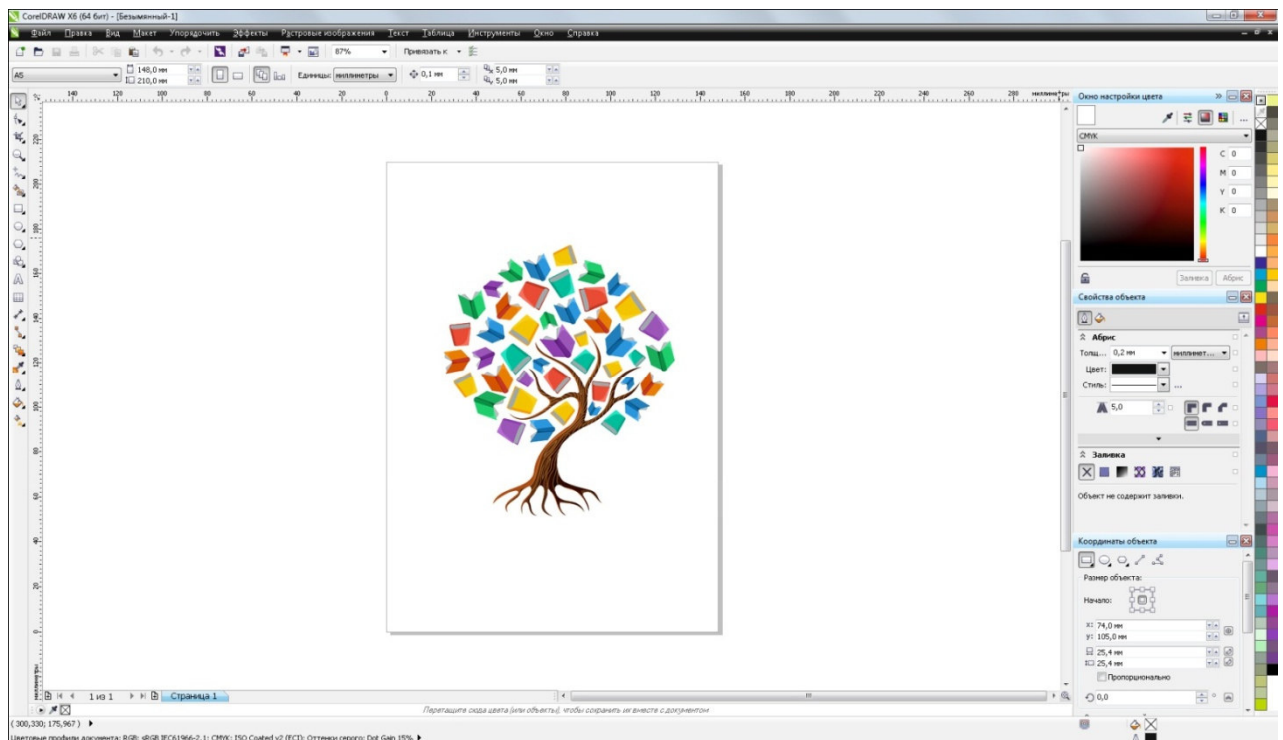


Рис. 2.6. Інтерфейс *CorelDRAW*

Для вирішення задачі вибору векторного графічного редактора порівнюємо програми: *Adobe Illustrator* та *CorelDRAW* (табл. 2.2) [69-70].

Порівняльний аналіз векторних графічних редакторів

Параметри		<i>Adobe Illustrator</i>	<i>CorelDRAW</i>
Вимоги до апаратного забезпечення	частота процесора, <i>GHz</i>	1.5	1.5
	оперативна пам'ять, <i>MB</i>	1000	2000
Кросплатформеність		<i>Windows, macOS</i>	<i>Windows</i>
Підтримка кольорових моделей		<i>RGB, CMYK, HSL, HSB, Lab, Greyscale</i>	<i>RGB, CMYK, HSL, HSB, Lab, Greyscale</i>
Можливість роботи з популярними форматами		Присутня	Присутня
Зручність інтерфейсу		Перенавантаженість панелі інструментів	Ергономічна панель інструментів
Функціональність інструментів		Велика кількість налаштувань	Велика кількість налаштувань
Можливості роботи з текстом		Повноцінні	Повноцінні
Швидкість роботи		Помірна	Висока
Налаштування експорту файлів		Наявне	Наявне
Ціна ліцензії, грн.		27000 (на 1 рік)	15000 (безтермінова)

Порівняльний аналіз програмного забезпечення для створення макетів сторінок та маніпуляції з текстовим та графічним наповненням проводився за наступними критеріями: швидкість роботи, зручність інтерфейсу, можливості роботи з текстом, функціональність інструментів, підтримка кольорових моделей, можливість роботи з популярними форматами, налаштування експорту файлів. Враховуючи всі проаналізовані показники, обрано графічний редактор *Corel DRAW*.

2.2. Структура взаємодії *front-end* та *back-end*

Ієрархічний поділ процесу розробки зумовлений складністю та багаторівневістю сучасних веб-ресурсів. Він складається із двох частин: клієнтська (*front-end*) та серверна (*back-end*) розробки.

Front-end – це створення користувацького інтерфейсу, функціональності та інтерактивності, що виконується на стороні клієнта веб-ресурсу або програми [50]. Щоб зрозуміти, що таке *front-end* розробка, необхідно відкрити сторінку будь-якого ресурсу та переглянути код даної сторінки в браузері. Цей код є прикладом роботи *front-end* розробки, що завантажується в браузер користувача, де відбувається його парсинг, і даний код можна побачити власними очима. Код сторінки описує все, що користувач бачить перед собою: кольори, верстку, шрифти, розташування графічних елементів тощо (рис. 2.7).



Рис. 2.7. Принцип роботи *front-end*

Сучасна *front-end* розробка тримається на трьох основних мовах: *HTML*, *CSS*, *JavaScript*. Дану структуру називають делегуванням відповідальності, де *HTML* відповідає за структуру, *CSS* за дизайн, а *JavaScript* за інтерактивність (рис. 2.8). Щоб веб-ресурс був більш досконалим, *front-end* розробник співпрацює з дизайнерами, програмістами та *UX*-аналітиками, для створення зручного та конкурентноспроможного продукту [50].

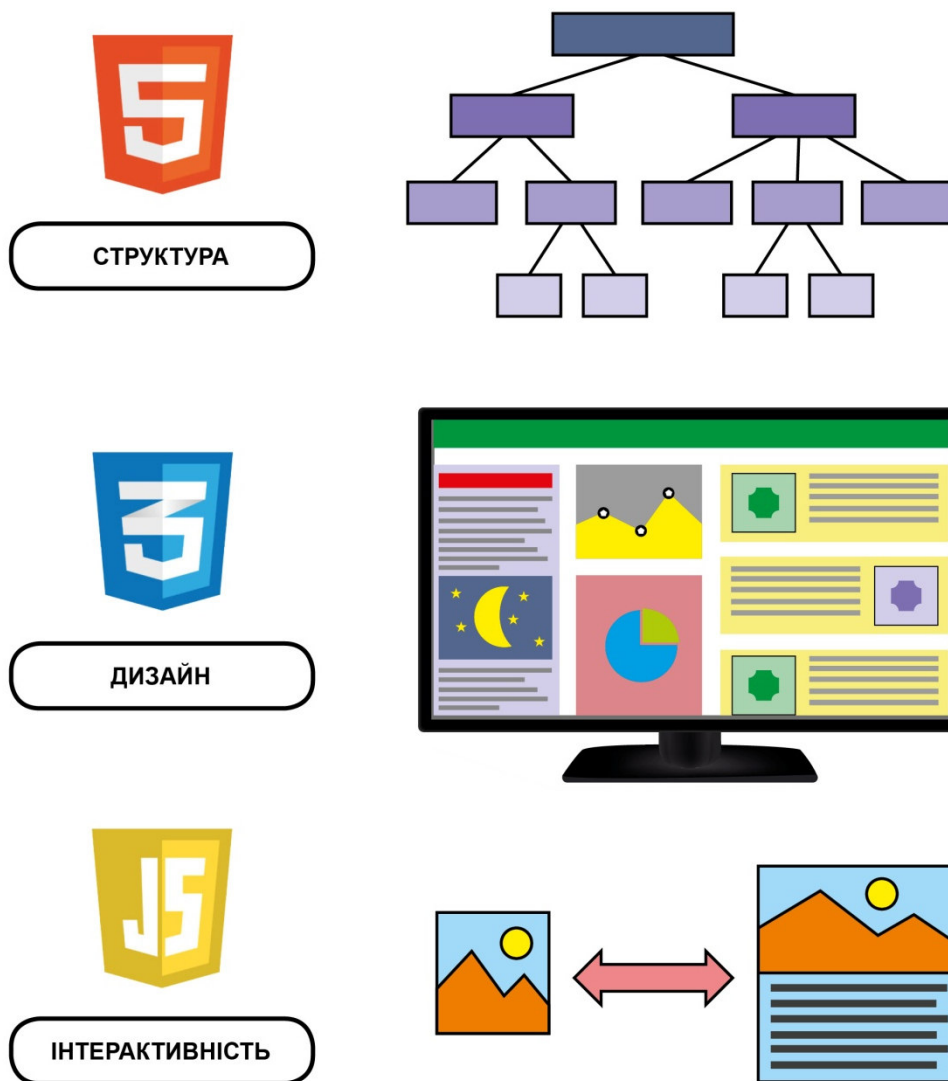


Рис. 2.8. Делегування відповідальності в *front-end*

HTML (HyperText Markup Language) – мова розмітки документів, за допомогою якої формується структура сторінки: заголовки, абзаци, списки тощо. Структура визначає елементи, необхідні для розміщення статичного або динамічного вмісту, а також є основною платформою для веб-ресурсів. Через розмаїття пристроїв з доступом в Інтернет та інтерфейсів для взаємодії з мережею такий базовий аспект, як структура, є надзвичайно важливою частиною документа, оскільки вона повинна забезпечувати форму, порядок та гнучкість. До структури документів *HTML* ставляться високі вимоги. Всі частини документа мають бути відокремленими одна від одної, кожна з них оголошена та укладена в певні теги. Тегами в *HTML* є ключові слова (оточені кутовими дужками), що можуть доповнюватися властивостями, відомими як атрибути чи *inline*-стилями. Більшість

тегів *HTML* парні, всередині яких знаходиться вміст, хоча й існує невелика кількість непарних тегів [12, с. 27-41].

Дуже важливою складовою в *HTML* верстці є семантика, яка відповідає за відповідність тегів до вмісту інформації, що знаходиться всередині них. Дотримуючись підходу із використанням семантики, можна гарантувати швидше завантаження сторінок та набагато якісніше ранжування пошуковими системами.

CSS (Cascading Style Sheets) – мова для опису і стилізації зовнішнього вигляду документа. Завдяки *CSS*-коду браузер розуміє, як саме відображати елементи. *CSS* задає кольори і параметри шрифтів, визначає, як будуть розташовуватися різні блоки веб-ресурсу тощо. *CSS* має доволі простий синтаксис, що складається із двох частин: селектора та блоку стилів, оточеного фігурними дужками. Селекторів існує надзвичайно багато, але застосовують, як правило, лише два основних – по класу і за ідентифікатором [14, с. 46-49].

Один *CSS*-файл містить інформацію про стиль всього веб-ресурсу, а це означає, що корегуючи лише одне правило можна кардинально змінити зовнішній вигляд. Сторінки завантажуються набагато швидше через те, що сучасні браузери використовують кешування для стилів та здатні їх застосовувати для всіх сторінок, а не до кожної окремо [14, с. 44-46].

JavaScript – це мова, що створювалася для того, щоб оживити веб-сторінки. Його завдання реагувати на дії користувача – обробляти кліки мишкою, переміщення курсора, натискання клавіш [26, с. 37]. Існує можливість посилення запитів на сервер та завантаження даних без перезавантаження сторінки, що дозволяє вводити повідомлення і багато іншого.

На *JavaScript* програмуються сценарії, які є набором інструкцій, яким повинен слідувати комп'ютер для досягнення тієї чи іншої мети. Сценарій, по своїй суті, це така сама програма, єдиною відмінністю якого є те, під керуванням якого середовища він виконуються. Сценарії на мові *JavaScript* інтерпретуються, але інтерпретація виконується не в явному вигляді, а засобами браузера [4, с. 10].

JavaScript допомагає поліпшити інтерактивність веб-сторінок за допомогою наступних можливостей [6, с. 10-11]:

– доступ до контенту (дозволяє виділити чи знайти на сторінці необхідний елемент з певним атрибутом або *id*, довідатися яку інформацію було введено в текстове поле тощо);

– зміна контенту (використовується для додавання на сторінку або для видалення з неї елементів, атрибутів і тексту);

– програмування правил (дозволяє вказати послідовність операцій, які повинен виконати браузер крок за кроком);

– реагування на події (надає можливість створити сценарій, підготовлений до запуску після конкретної події).

Back-end розробка – це набір апаратно-програмних засобів, за допомогою яких реалізується логіка роботи ресурсу. Тобто, це те, що приховано від очей користувача і відбувається поза його браузером та комп'ютером. Яскравий приклад *back-end* розробки: коли користувач вводить запит на сторінці пошуковика та натискає клавішу *Enter*, робота *front-end* закінчується і починається *back-end*. Запит відправляється на сервер, де розташовані алгоритми пошуку. Сервер – це більш потужний комп'ютер, що виконує певні функції. Він зберігає дані та відповідає на запити користувачів. Щойно на моніторі з'являється інформація, яку шукав користувач, знову відбувається повернення в зону *front-end*. По суті, *back-end* – це процес об'єднання сервера з користувачем за допомогою баз даних, *API* та операційних систем [71].

Back-end розробник може застосовувати різноманітні інструменти, що доступні на сервері. Як правило, в його розпорядженні наявні розмаїті мови програмування, зокрема *PHP*, *Java* та *Python*. Також для *back-end* розробки використовуються різні системи управління базами даних: *MySQL*, *MongoDB*, *Cassandra* тощо. Самою поширеною комбінацією в *back-end* розробці є поєднання *PHP* з *MySQL* (рис. 2.9). Залежно від цілей, *back-end* розробник може виконувати наступні функції: забезпечувати кібербезпеку ресурсу та даних користувачів, створювати та інтегрувати бази даних, налаштовувати технології резервного копіювання та відновлення [71].

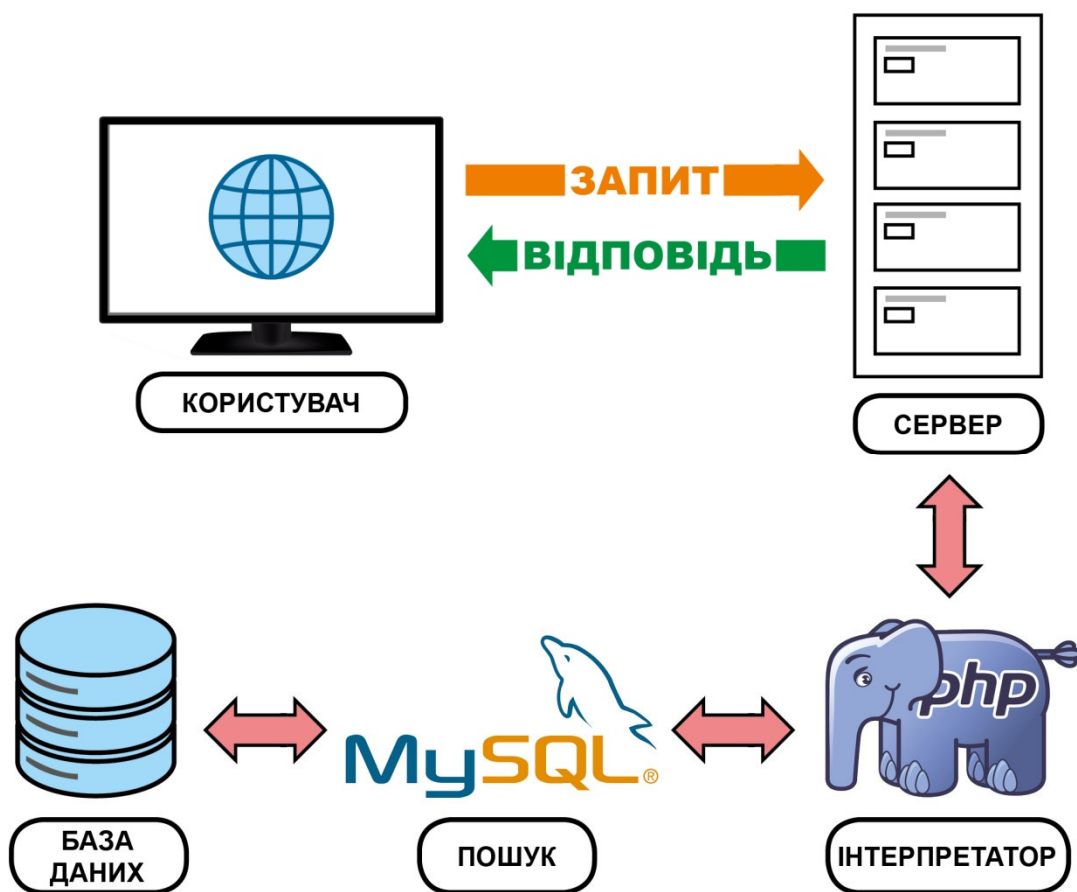


Рис. 2.9. Делегування відповідальності в *back-end*

PHP (*Hypertext Preprocessor*) – мова, що застосовується у сфері веб-розробок, головна ціль якої генерація *HTML*-сторінок на стороні веб-сервера. Схема опрацювання *PHP* коду наступна, коли користувач завантажує певну сторінку, браузер надсилає запит до сервера, в якому відбувається виконання *PHP* коду за допомогою спеціального інтерпретатора з подальшою генерацією *HTML* коду та відправлення його назад користувачу [9, с. 124-126]. Важливо відмітити, що користувач не в змозі побачити *PHP* код, оскільки він потрапляє до браузера опрацьований інтерпретатором у вигляді *HTML* коду.

В мові *PHP* використовуються сесії, що є простим способом зберігання інформації для окремих користувачів з унікальним ідентифікатором сесії. Як правило, їх застосовують для збереження стану між запитами сторінок. Відсутність ідентифікатора сесії або сесійного *cookie* повідомляє *PHP* про те, що необхідно створити нову сесію і згенерувати новий ідентифікатор сесії. Оскільки сесії

зберігаються на стороні сервера, а не на стороні клієнта, то вони набагато краще захищені від хакерів через те, що їх майже неможливо підробити [9, с. 622-624].

MySQL – це мова, основна ціль якої надання можливості створення простих запитів до бази даних за допомогою команд. База даних є структурованою колекцією записів або даних, що зберігаються в комп'ютерній системі і організовані так, що можна швидко здійснювати пошук і витягти необхідну інформації. У базі даних *MySQL* є одна або декілька таблиць, кожна з яких складається з записів або рядків. У середині рядків знаходяться різні стовпці або поля, в яких і містяться дані [19, с. 192-193].

Мова *MySQL* є універсальною, оскільки використовується в усіх операційних системах та має драйвери для будь-якої мови програмування. Вона використовується багатьма розробниками завдяки простоті інсталяції та налаштуванні, а наявність безлічі плагінів і допоміжних додатків спрощує роботу. Система *MySQL* володіє практично всім необхідним інструментарієм, який може знадобитися в роботі як з малими, так і з великими об'ємами даних [58].

Взаємодія між *front-end* та *back-end* відбувається за допомогою наступної структури (рис. 2.10): *front-end* відправляє призначену для користувача інформацію в *back-end*, де вона опрацьовується і повертається назад, прийнявши зрозумілу форму. Робота *front-end* починається зі звернення до веб-клієнта, який в свою чергу обробляє наступні складові інформації, що включають візуальне відображення, репозиторій даних та інтерактивність. Візуальне відображення відповідає за структуру та дизайн і складається з *HTML* та *CSS* (фреймворки та препроцесори). Репозиторій даних зберігає інформацію трьох видів: локальну, сесійну та кеш. За інтерактивність відповідає *JavaScript* з можливістю застосування фреймворків та компіляторів. Робота *back-end* починається після того, як веб-клієнт звертається до веб-сервера, який в свою чергу відповідає за репозиторій даних, мови генерації, кешування та інтеграцію додатків. Репозиторій даних відповідає за запити до бази даних за допомогою однієї з систем керування: *MySQL*, *Oracle* чи *MongoDB*. Мови генерації формують контент за допомогою однієї з мов: *PHP*, *Python*, *Java* чи *JavaScript*. Інтеграція додатків відповідає за *API*, що включає в себе *REST* або *AJAX*.

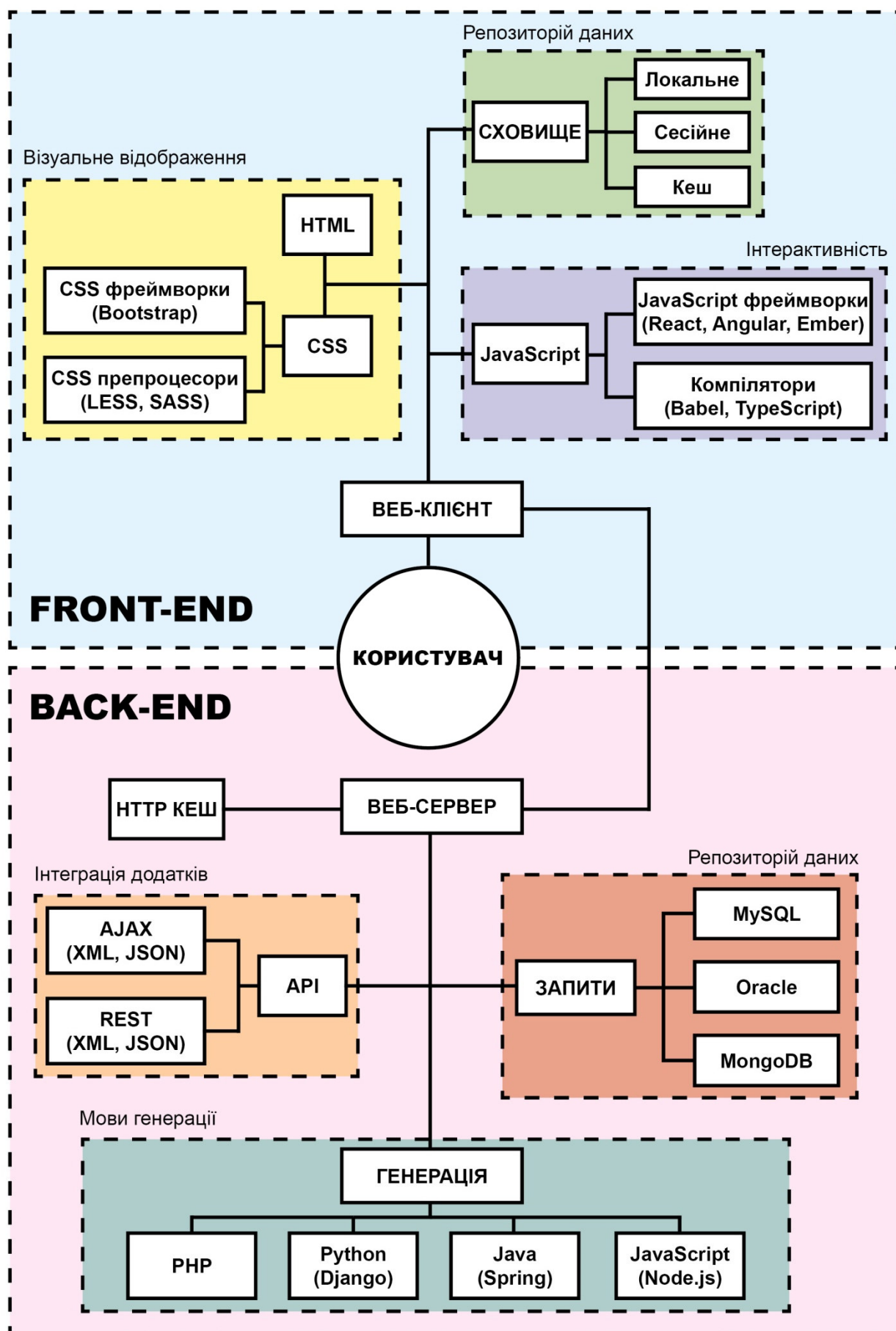


Рис. 2.10. Структура взаємодії front-end та back-end

Існують наступні популярні варіанти взаємодії *front-end* і *back-end* [16]:

- *HTTP*-запит безпосередньо надсилається на сервер, потім сервер шукає інформацію, вбудовує її в шаблон і повертає у вигляді *HTML*-сторінки;
- варіант з використанням інструментарію *AJAX* (*Asynchronous JavaScript and XML*), де запит відправляє *JavaScript*, завантажений в браузер, а відповідь приходить в форматі *XML* або *JSON*;
- односторінкові додатки, які завантажують дані без оновлення сторінки (реалізується за допомогою *AJAX* або фреймворків *Angular* і *Ember*);
- *Ember* або бібліотека *React* допомагають використовувати додаток і на сервері, і в клієнті (*front-end* і *back-end* взаємодіють через *AJAX* і *HTML*-код, який обробляється на сервері).

2.3. Вибір системи керування веб-ресурсом

В сучасній розробці веб-ресурсів для спрощення та пришвидшення процесу створення все частіше звертаються до спеціальних систем, відомих як *CMS*. Система керування вмістом (*CMS*) – це комп'ютерна система або програма, яка дозволяє публікувати, редагувати або змінювати вміст, а також надає можливість модифікувати компоненти веб-ресурса. Застосування *CMS* обумовлено завдяки наступним можливостям [53, с. 11-13]:

- динамічний вміст (миттєве відображення внесених змін);
- інтуїтивне редагування (можливість зміни наповнення, що не потребує знань в сфері програмування та верстки);
- ефективне керування структурою (можливість додавати або переміщувати в потрібне місце веб-сторінки);
- керування дизайном (налаштування дизайну з використанням шаблонів);
- використання мультимедійних елементів (просте додавання фото, відео, аудіо завдяки вбудованому інструментарію);
- використання рівнів доступу (розподілення можливостей редагування ресурсу).

Процес вибору системи керування є важливим фактором для виконання завдання дипломної роботи, оскільки від цього буде залежати зручність створення, наповнення контентом та користування веб-ресурсом. Під час вибору *CMS* необхідно враховувати наступні основні критерії:

- зручність користування (функціональний візуальний інтерфейс з можливістю швидкого додавання та редагування контенту);
- доступність покращення функціональних можливостей (наявність модифікуючих плагінів);
- дизайнерські можливості (можливість встановлення різноманітних тем та їх подальша модернізація);
- оновлюваність (частота апгрейду *CMS* та додатків);
- можливості оптимізації пошукових систем (підтримка принципів *SEO*);
- швидкодія (швидкість завантаження та навантаження на хостинг).

В. Морріс (*Will Morris*) [57] досліджуючи сферу популярних систем керування вмістом, сформував перелік найбільш зручних та популярних. Для аналізу та вибору *CMS* обираємо трійку лідерів із списку: *WordPress*, *Joomla* та *Drupal*.

WordPress – це потужна семантична видавнича платформа написана за допомогою *PHP* та *MySQL*, що оснащена набором функцій, покликаних суттєво спростити процес створення веб-ресурсу. *WordPress* має простий інтерфейс, подібний до офісного програмного забезпечення, а отже не вимагає додаткового процесу вивчення та досвіду користування (рис. 2.11). Надає можливість зручного додавання мультимедійного контенту завдяки завантажувачу файлів із перетягуванням. Передбачена функція збереження написаного контенту із можливістю повернення до попередньої версії. Надає змогу планувати публікації через деякий час або дозволяє оновлювати матеріал, що був розміщений раніше. *WordPress* організовує вміст за днями, місяцями, роками, авторами, категоріями та динамічно створює архіви, які можна переглянути за обраним параметром. Кожна *URL*-адреса інтуїтивно написана для користувачів і описує те, що включає вміст, а не там, де він дислокується в базі даних. *WordPress* повністю задовільняє стандарти *W3C*, а значить гарантує повну сумісність з усіма браузерами [72].

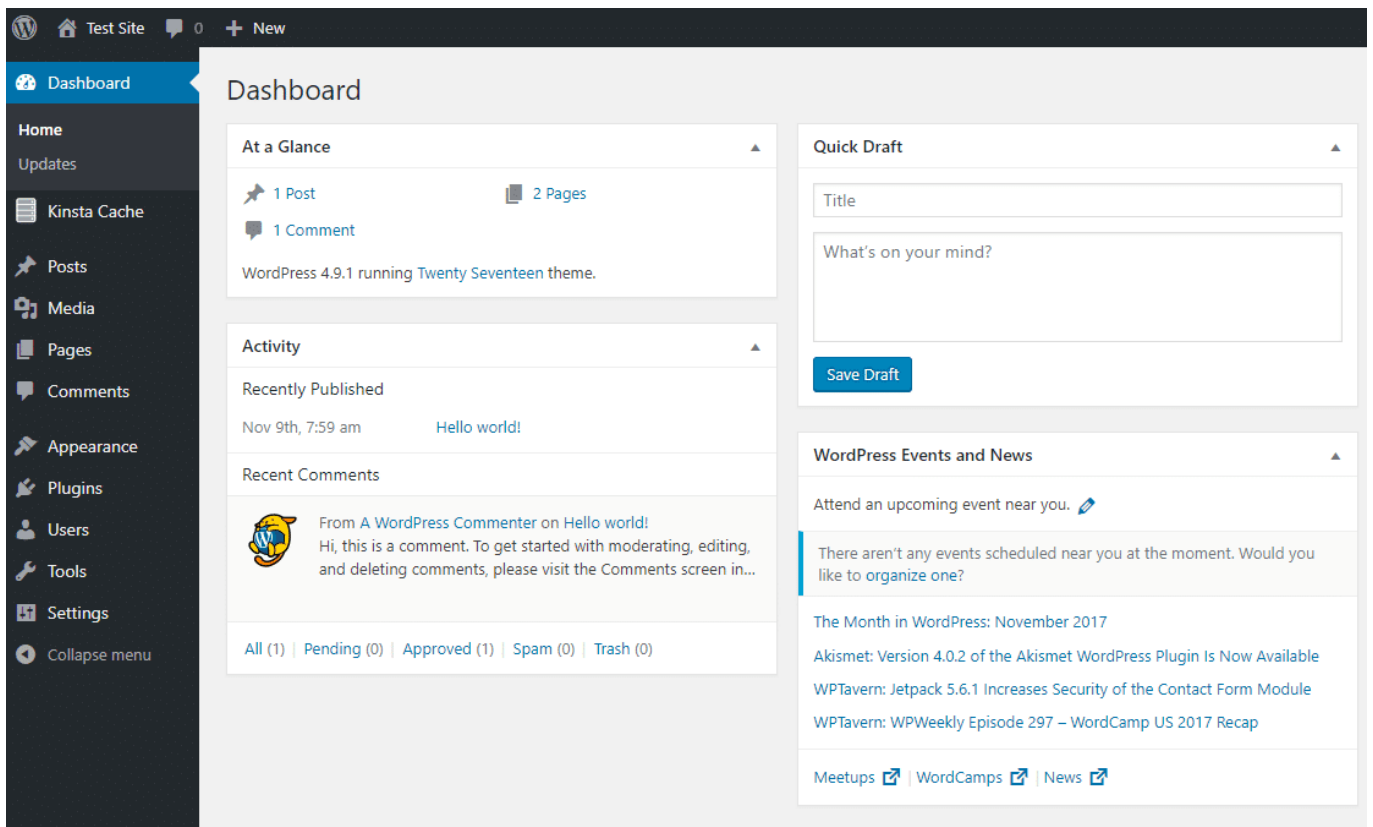


Рис. 2.11. Інтерфейс *WordPress*

Joomla – це програмний пакет написаний за допомогою *PHP*, що використовується для створення, організації, управління та публікації вмісту для веб-ресурсів. За допомогою зручного інтерфейсу, який контролює всі функціональні можливості, користувачі можуть створювати вміст, що легко публікується та редагується (рис. 2.12). За допомогою *Joomla* є можливість оптимізувати заголовки, описи статей, мультимедійні файли для того, щоб покращити *SEO*-оптимізацію та рейтинг веб-ресурсу в пошукових системах. Вбудована технологія *Bootstrap* гарантує коректне відображення інтерфейсу розробника та адаптивність сторінок абсолютно на всіх типах пристроїв. Менеджер додаткових пакетів дозволяє вдосконалити функціональні можливості веб-ресурсу завдяки інсталяції плагінів та модулів. Бібліотека шаблонів спрощує дизайнерське оформлення веб-ресурсу та надає можливість розробнику вносити певні зміни в макет. Вбудований оновник передбачає автоматизовану перевірку, який інформує про необхідність оновити компоненти. В *Joomla* інтегровано стиснення з використанням технології *Gzip*, що підвищує швидкість та продуктивність веб-ресурсу [46].

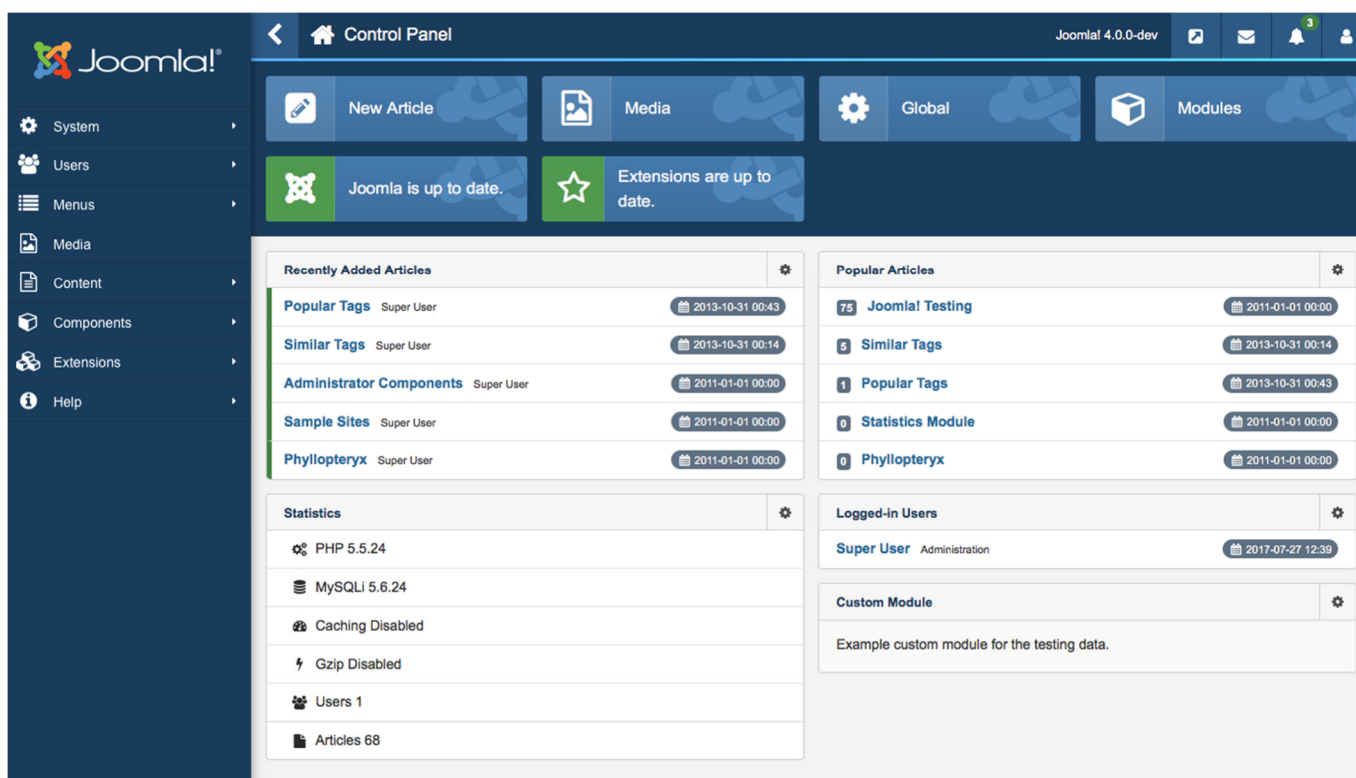


Рис. 2.12. Інтерфейс *Joomla*

Drupal – це платформа з відкритим кодом написана за допомогою *PHP*, що використовується для створення веб-ресурсів будь-якої складності. Інтерфейс пропонує велику кількість інструментів та налаштувань для створення веб-ресурсу (рис. 2.13), а також включає візуальний редактор, що працює за технологією *WYSIWYG*. *Drupal* дозволяє повністю контролювати повноваження (шляхом розподілу на ролі) користувачів для доступу до різних функцій веб-ресурсу. Веб-ресурс створений за допомогою *Drupal* буде оптимально налаштований для пошукових систем за допомогою надання коректних *URL*-адрес, постійних посилань на вміст, налаштовуваних метаданих сторінок та автоматично генерованої *XML*-карти ресурсу. Велика кількість створених тем (шаблонів) дозволяє повністю налаштувати дизайн та макет веб-ресурсу, який відповідає стандартам *HTML* та *CSS*. За замовчуванням *Drupal* має вбудовані функції для відстеження дій користувачів та збору статистичних даних. В *CMS* інтегровано спеціальні додатки, які відповідають за кешування даних та *SEO*-оптимізацію, що дозволяють завантажувати веб-ресурси максимально швидко. Для виявлення технічних проблем в *Drupal* передбачено модуль, який сигналізує про помилки та вказує шляхи для їх вирішення [39].

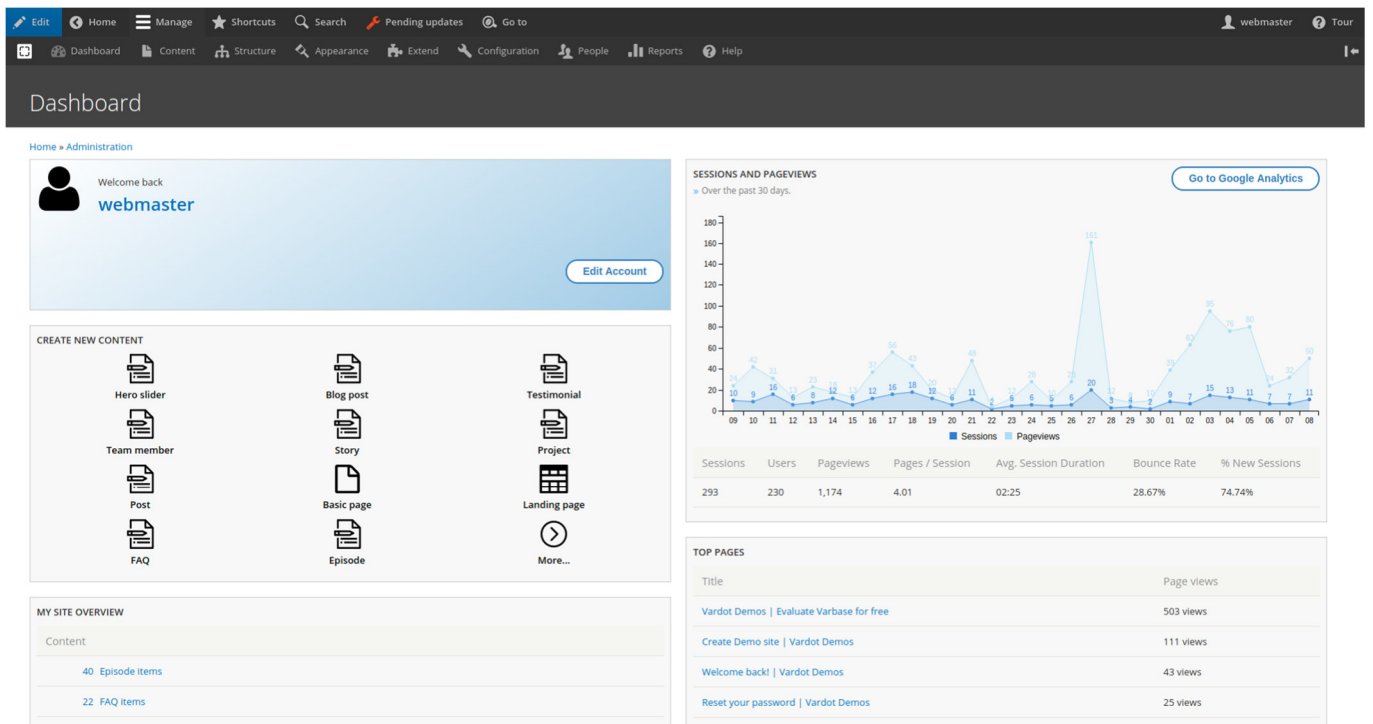


Рис. 2.13. Інтерфейс *Drupal*

Для вирішення задачі вибору системи керування вмістом порівнюємо три *CMS: WordPress, Joomla* та *Drupal* (табл. 2.3) [39, 46, 72].

Таблиця 2.3

Порівняльний аналіз систем керування вмістом

Параметри	<i>WordPress</i>	<i>Joomla</i>	<i>Drupal</i>
1	2	3	4
Мова програмування	<i>PHP</i>	<i>PHP</i>	<i>PHP</i>
База даних	<i>MySQL</i>	<i>MySQL</i>	<i>MySQL, PostgreSQL</i>
Веб-сервер	<i>Apache</i>	<i>Apache</i>	<i>Apache, IIS, nginx</i>
Підтримка <i>UTF-8</i>	Повна	Часткова	Повна
Кількість локалізацій, мови	62	66	100
Зручність інсталяції	Встановлення одним кліком	Встановлення одним кліком	Встановлення одним кліком
Призначення	Блоги, веб-ресурси, портали, форуми	Комерційні ресурси, соціальні мережі	Блоги, веб-ресурси, портали, форуми

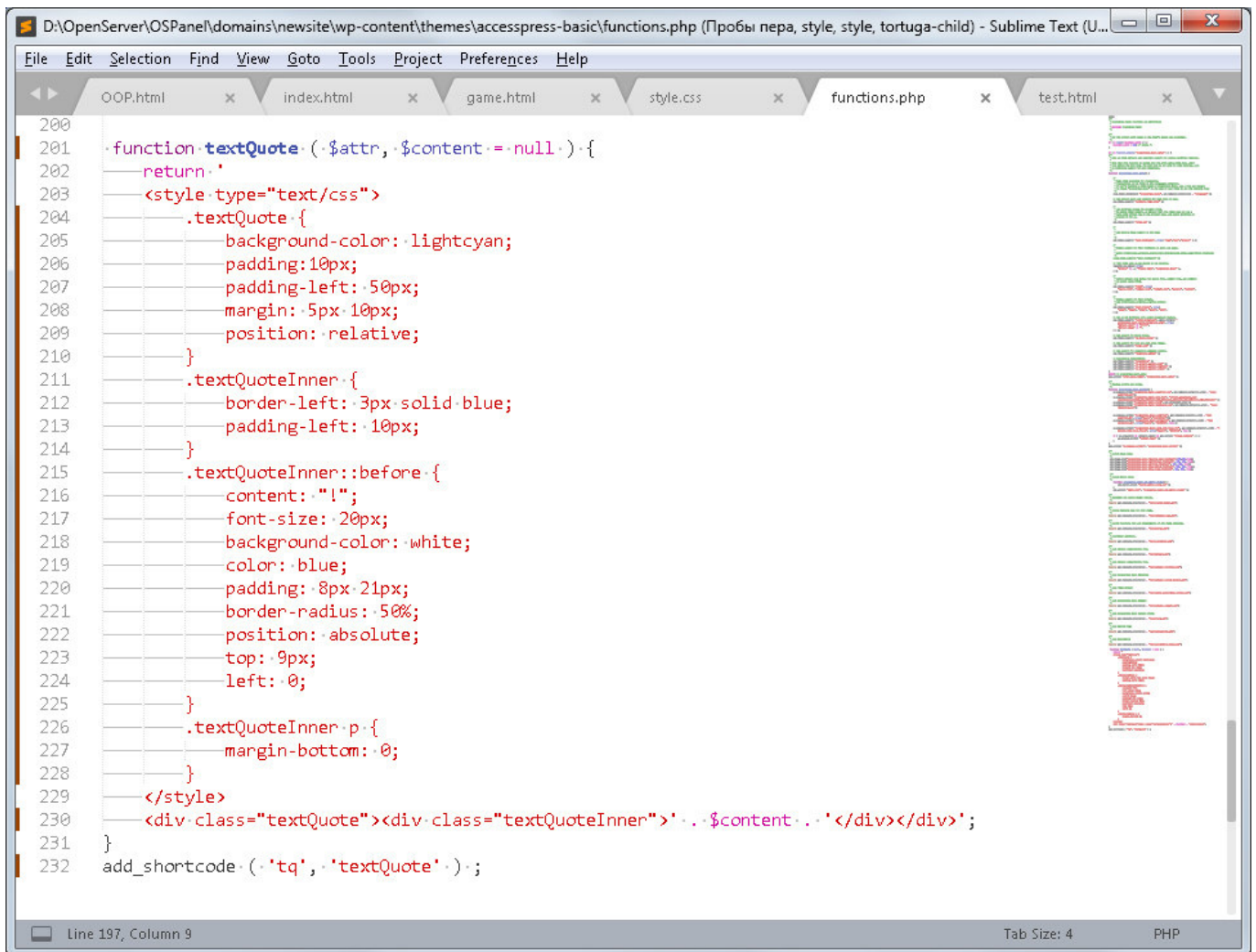
1	2	3	4
Зручність користування	Гнучкий інтерфейс, що не вимагає навичок веб-розробки	Функціональний інтерфейс, що вимагає певних навичок	Інтерфейс з потужними можливостями, але складний в опануванні
Доступність покращення	Наявно понад п'ятдесят п'ять тисяч плагінів	Наявно понад тридцять дві тисячі плагінів	Наявно понад семи тисяч плагінів
Дизайнерські можливості	Існує понад дві тисячі шаблонів	Існує понад одну тисячу шаблонів	Існує понад дві тисячі шаблонів
Макетування	Обмежено можливостями шаблону	Можливість розширення шаблону	Обмежено можливостями шаблону
<i>WYSIWYG</i> редактор	Наявний	Наявний	Наявний
Оновлюваність	Висока частота апгрейду <i>CMS</i> та додатків	Висока частота апгрейду <i>CMS</i> та додатків	Висока частота апгрейду <i>CMS</i> та додатків
Можливості оптимізації пошукових систем	Максимальний інструментарій для покращення <i>SEO</i>	Мінімальний інструментарій для покращення <i>SEO</i>	Достатній інструментарій для покращення <i>SEO</i>
Кешування сторінок	Присутнє	Присутнє	Присутнє
Швидкодія	Оптимальні можливості для подолання веб-трафіку, але вони залежать від можливостей хостинг-сервісу	Для оптимізації швидкодії необхідно постійно коректувати налаштування	Найкращі показники завантаження та навантаження на хостинг
Пошук по веб-ресурсу	Присутній	Присутній	Присутній
Адаптивність	Наявна	Наявна	Наявна
Багатомовність	Присутня	Присутня	Присутня
Можливість модернізації	Система з відкритим кодом	Система з відкритим кодом	Система з відкритим кодом
Ліценція	Безкоштовна	Безкоштовна	Безкоштовна

З проведеного огляду та порівняльного аналізу видно, що всі розглянуті системи керування вмістом володіють необхідним функціоналом для створення веб-ресурсів, але вимагають різного рівня навичок у сфері веб-розробок. Для виконання дипломної роботи було обрано *CMS WordPress*. На користь вибору вплинули наступні фактори: зручний інтерфейс, найбільша кількість плагінів та шаблонів, оновлюваність системи, *SEO*-оптимізація.

Доповнюємо базову комплектацію *WordPress* наступними плагінами для підвищення функціональних можливостей *CMS* [41, 65]:

- *Yoast SEO* (дозволяє точніше налаштовувати параметри оптимізації: мета-описів, медіа-файлів та ключових слів);
- *Classic Editor* (застосовує панель інструментів, подібну до офісного програмного забезпечення);
- *TinyMCE Advanced* (доповнює функціонал візуального редактора);
- *WordFence Security* (слідкує за кібер-безпекою веб-ресурсу за допомогою *firewall* та сканує веб-ресурс на вміст шкідливих файлів);
- *WP Google Maps* (надає можливість використання *Google*-мапи на сторінках веб-ресурсу);
- *Google Analytics* (відстежує відвідування ресурсу користувачами, час перегляду сторінок та іншу статистичну інформацію);
- *Smush* (оптимізує ілюстраційний матеріал з метою пришвидшення завантаження веб-ресурсу та економії місця у файловому сховищі);
- *Broken Link Checker* (ідентифікує розбиті посилання, які більше не працюють, відсутні зображення або будь-які переадресації).

Подальша модернізація системи керування вмістом *WordPress* включає написання шорткодів, що слугують для застосування будь-якої функції до обраного фрагменту веб-ресурсу для молодосвідчених веб-розробників. Шорткод створюється в *CMS WordPress* у файлі *functions.php* за допомогою *PHP*-коду та складається із трьох компонентів: ініціалізації, аббревіатури виклику та функції для виконання. Пишемо шорткоди для візуального виділення фрагментів із сенсовим навантаженням для економії часу на заздалегідь розроблених функціях (рис. 2.14).



```
200
201 function textQuote ( $attr, $content = null ) {
202     return '
203     <style type="text/css">
204         .textQuote {
205             background-color: lightcyan;
206             padding: 10px;
207             padding-left: 50px;
208             margin: 5px 10px;
209             position: relative;
210         }
211         .textQuoteInner {
212             border-left: 3px solid blue;
213             padding-left: 10px;
214         }
215         .textQuoteInner::before {
216             content: "!";
217             font-size: 20px;
218             background-color: white;
219             color: blue;
220             padding: 8px 21px;
221             border-radius: 50%;
222             position: absolute;
223             top: 9px;
224             left: 0;
225         }
226         .textQuoteInner p {
227             margin-bottom: 0;
228         }
229     </style>
230     <div class="textQuote"><div class="textQuoteInner">' . $content . '</div></div>';
231 }
232 add_shortcode ( 'tq', 'textQuote' );
```

Рис. 2.14. Шорткод для виділення важливого тексту

Порівняльний аналіз систем керування вмістом проводився за наступними ключовими критеріями: зручність користування, доступність покращення, дизайнерські можливості, оновлюваність, можливості оптимізації пошукових систем та швидкодія. На основі цього було обрано *CMS WordPress* із подальшим вдосконаленням його функціональних можливостей за допомогою плагінів та написання шорткодів.

2.4. Етапи технології створення веб-ресурсу

Розробка та реалізація веб-ресурсу являє собою тривалий та трудомісткий процес, що потребує професіоналізму та відповідності у виконанні етапів технологічного процесу.

У якості робочої технології створення веб-ресурсу, за основу обрано технологію запропоновану Д. Сітбоном (*Jonathan Sitbon*) [66] із подальшим вдосконаленням, розширенням та уточненням етапів (рис. 2.15).



Рис. 2.15. Етапи технології створення веб-ресурсу

Першим етапом є написання технічного завдання в якому визначаються цілі та завдання веб-ресурсу, що опираються на наступні фактори: мета створення, тематика, цільова аудиторія, функціональні можливості. Особливу увагу слід приділити технічним особливостям веб-ресурсу, зокрема адаптивності (необхідність відображення на пристроях з будь-яким розміром екрану) та кросплатформеності (перелік браузерів та їх версій для підтримки). Технічне завдання описує вид структури, кількість сторінок та взаємозв'язок між ними. В описі сторінок вказується наявність структурних елементів (хедер, тіло, сайдбар, футер) з компонентами (логотип, меню, контактна інформація, пошук, віджети, інформаційні

блоки). Також описуються вимоги до дизайнерського оформлення, а саме бажану кольорову гаму, основний тип шрифту (із засічками, без засічок) та можливість застосування акцентуючих елементів.

Другий етап включає в себе вибір хостингу та доменного імені. На вибір хостингу впливають такі фактори, як обсяг зберігаємої інформації, функціональність, програмна підтримка, цінова категорія, наявність *SSL*-сертифікату. Виходячи з цих параметрів є можливість вибору з чотирьох варіантів: віртуальний хостинг, віртуальний сервер, хмарний хостинг, виділений сервер. Від вибору доменного імені залежить сприйняття користувачами змістовного наповнення, а також підвищення *SEO*-параметрів в середовищі пошукових систем. Оптимальне доменне ім'я має враховувати наступні критерії [66]:

- стислість (довгі назви сприяють підвищенню помилок при друкуванні *URL*-адрес);
- простота (уникнення спеціальних символів та цифр);
- ідентифікованість (має містити інформативну назву та вказувати на власника).

На третьому етапі обирається система керування вмістом. Під час вибору необхідно зважати на наступні критерії: мову програмування, базу даних, веб-сервер, локалізацію, метод інсталяції, призначення веб-ресурсу, зручність користування, модернізація плагінами, можливості маніпуляції з шаблоном, частота оновлення *CMS* та модулів, наявність вбудованої *SEO*-оптимізації, кешування даних, швидкодія, адаптивність, цінова політика.

На четвертому етапі відбувається розробка дизайнерського оформлення веб-ресурсу на основі технічного завдання та формується унікальний візуальний стиль. Веб-дизайн повинен бути збалансованим, містити унікальне наповнення (без застосування текстового та ілюстративного матеріалу наявного в Інтернеті), без надмірного використання візуальних інтерактивних ефектів. При створенні макету важливо дотримуватися *UX/UI* технології. *UI (User Interface)* відповідає за зовнішній вигляд (інтерфейс) із висунутим вимог до створення декоративних елементів (піктограм, які сприяють покращенню сприйняття інформації), органів керування

(кнопок, різноманітних повзунків), відображення матеріалу (розбиття елементів за певною структурною сіткою) та всіх інших елементів, з якими користувачу необхідно взаємодіяти. *UX (User Experience)* відповідає за зручність (наскільки структура відповідає логіці та чи є вона інтуїтивно зрозумілою) користування веб-ресурсом. Підбирається кольорова гама, що формує емоційну реакцію користувача, яка керується основними принципами теорії кольору (гармонічне поєднання кольорів, оптимальна кількість кольорів дорівнює трьом). Обирається шрифтове оформлення для тексту, яке забезпечить зручність сприйняття текстової інформації. До параметрів налаштування шрифтів відносяться: накреслення, розмір, інтерліньяж, кернінг, трекінг, абзацні відступи тощо. Доцільно застосовувати не більше трьох різновидів шрифтів для оформлення матеріалу: заголовків та анонсів, основного тексту, акцентування на важливу інформацію. Веб-дизайн необхідно створювати спираючись на принципи рівноваги, ефективності, естетичності, гармонії, передбачуваності, прозорості, лаконічності, орієнтованості на користувача та акцентування. Якщо в технічному завданні передбачується підтримка адаптивності, то розроблюються макети веб-ресурсу під необхідні розміри екранів.

Верстка та програмування виконуються на п'ятому етапі технологічного процесу для реалізації концептуальної дизайнерської моделі у відповідності до вимог технічного завдання. На обраний хостинг встановлюється система керування вмістом, яка під'єднується до бази даних. Під час процесу створення веб-ресурсу його експортують та підключають до локального серверу з метою покращення швидкості та зручності всіх необхідних маніпуляцій, оскільки стандартний редактор для написання коду доволі не функціональний. Обирається шаблон, який максимально задовольняє поставленим вимогам та виконуються його базові налаштування. Після цього відбувається модернізація шляхом створення дочірньої теми, в якій підключаються необхідні шрифти та виконується більш точне налаштування дизайну (позиціонування блоків та задання їм розмірів, налаштування кольорового оформлення, додавання псевдо-елементів та псевдо-класів). Інсталюються плагіни для забезпечення функціональних можливостей веб-ресурсу, описаних в технічному завданні та дизайнерській концепції. Програмуються

відсутні структурні елементи та вдосконалюються наявні можливості ресурсу, шляхом написання коду на відповідній мові. За необхідності корегуються правила адаптивності задля забезпечення коректного відображення на всіх розмірах екранів.

Наповнення контентом веб-ресурсу відбувається на шостому етапі, який передбачає розміщення текстового матеріалу на відповідних сторінках, а також підбирається та оброблюється ілюстративний матеріал, що слугує для доповнення та пояснення текстового контенту [66].

Сьомий етап стосується *SEO*-оптимізації та покращенню рейтингу веб-сторінок для пошукових систем. Проводиться оптимізація графічного вмісту та програмних файлів задля пришвидшення завантаження сторінки, оскільки швидкість являється одним із головним критерієм рейтингу *SEO*. Для ранжування налаштовуються *URL*-адреси та файл *robots.txt*, що повідомляє пошуковій системі про те, яким чином і де шукати інформацію розташовану на веб-ресурсі. Перевіряються метаописи (не більше ста шісттидесяти символів та інформативність викладення), правильність виставлення ключових слів (відповідність наповнення проставленим ключовим словам), довжина вмісту (контент, як мінімум, повинен містити понад дві тисячі слів), альтернативні описи для зображень та налаштованість посилань.

Тестування проводиться на восьмому етапі з метою перевірити функціонал веб-ресурсу у відповідність до зазначеного у технічному завданні. Перевіряється адаптивність відображення вмісту та структурних елементів на пристроях із різними розмірами екрану. Проводиться перевірка коректності відображення веб-ресурсу браузерами та підтримки запрограмованих функціональних можливостей. Тестується ергономічні складові ресурсу, що включають в себе зручність навігації, шрифти, розташування контенту, кольорова гама. Виявляються помилки в роботі пошукового механізму, а саме правильність знаходження необхідної інформації та наявність сторінки помилки.

На фінальному етапі відбувається публікація веб-ресурсу в мережі Інтернет, що відкриває доступ для всіх користувачів, а також надає можливості пошуковим системам індексувати даний ресурс.

Висновки до розділу

Порівняльний аналіз сучасного програмного забезпечення для написання коду проводився за наступними основними критеріями: наявність підсвічування синтаксису, можливість автоматичних відступів та автодоповнення коду, персоналізація інтерфейсу, здатність розділення на робочі зони, відображення міні-мапи, інтеграція модулів та додатків. На основі цього обрано текстовий редактор *Sublime Text 3* із подальшою його модернізацією шляхом залучення функціональних плагінів та написанням сніпсетів для додавання часто вживаних блоків коду. Порівняльний аналіз програмного забезпечення для створення макетів сторінок та маніпуляції з текстовим та графічним наповненням проводився за наступними критеріями: швидкість роботи, зручність інтерфейсу, можливості роботи з текстом, функціональність інструментів, підтримка кольорових моделей, можливість роботи з популярними форматами, налаштування експорту файлів. Враховуючи всі проаналізовані показники, обрано графічний редактор *Corel DRAW*.

Під час розгляду структури взаємодії *front-end* та *back-end* виявлено делегування відповідальності мов та обрано варіант, де *HTTP*-запит надсилається на сервер, який шукає інформацію та повертає її у вигляді *HTML*-сторінки.

Вибір системи керування вмістом виконувався за наступними ключовими критеріями: зручність користування, доступність покращення, дизайнерські можливості, оновлюваність, можливості оптимізації пошукових систем та швидкодія. На основі цього обрано *CMS WordPress* із подальшим вдосконаленням його функціональних можливостей за допомогою плагінів та написання шорткодів для візуального виділення фрагментів із сенсовим навантаженням.

Визначено етапи технології створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”: формування технічного завдання; вибір хостингу та доменного імені; вибір *CMS*; розробка дизайнерської концепції; верстка та програмування; наповнення контентом; *SEO*-оптимізація; тестування; публікація.

РОЗДІЛ 3
ПРОЕКТНА РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ СТВОРЕННЯ ВЕБ-РЕСУРСУ
„КАФЕДРА СУРДОПЕДАГОГІКИ ТА СУРДОПСИХОЛОГІЇ
ІМЕНІ М.Д. ЯРМАЧЕНКА НАЦІОНАЛЬНОГО ПЕДАГОГІЧНОГО
УНІВЕРСИТЕТУ ІМЕНІ М.П. ДРАГОМАНОВА”

3.1. Інформаційно-концептуальна модель веб-ресурсу

Інформаційно-концептуальна модель формує основні параметри майбутнього веб-ресурсу, що включає критерії підбору та розташування матеріалу, а також сприяє максимальному донесенню інформації до користувача. Основою для створення веб-ресурсу слугувала інформація, яку надала кафедра, що включала текстовий та ілюстративний матеріал. Замовником сформульовано задачі, які має виконувати веб-ресурс:

- інформаційна (надання актуальної інформації про діяльність кафедри);
- комунікативна (обмін інформацією представників кафедри з іншими користувачами);
- методична (ознайомлення з процесом, формами та методиками навчання, а також науковою та дослідницькою діяльністю).

При створенні структури веб-ресурсу, яка розділяється на внутрішню та зовнішню, формується перелік сторінок та взаємозв'язок між ними з урахуванням зручності орієнтації та пошуку інформації.

Будуємо внутрішню структуру веб-ресурсу, що включає в себе весь контент розбитий на основні теми (розділи), яка враховує продумані логічні зв'язки між ними (рис. 3.1).

Створюємо зовнішню структуру веб-ресурсу, яка відповідає за розташування основних важливих елементів на сторінці, що включає в себе логотип, меню, пошук, інформаційні блоки, контактну інформацію тощо (рис. 3.2). Опираючись на вимоги ергономіки, розміщуємо базові елементи в інтуїтивно зрозумілих місцях для користувача.

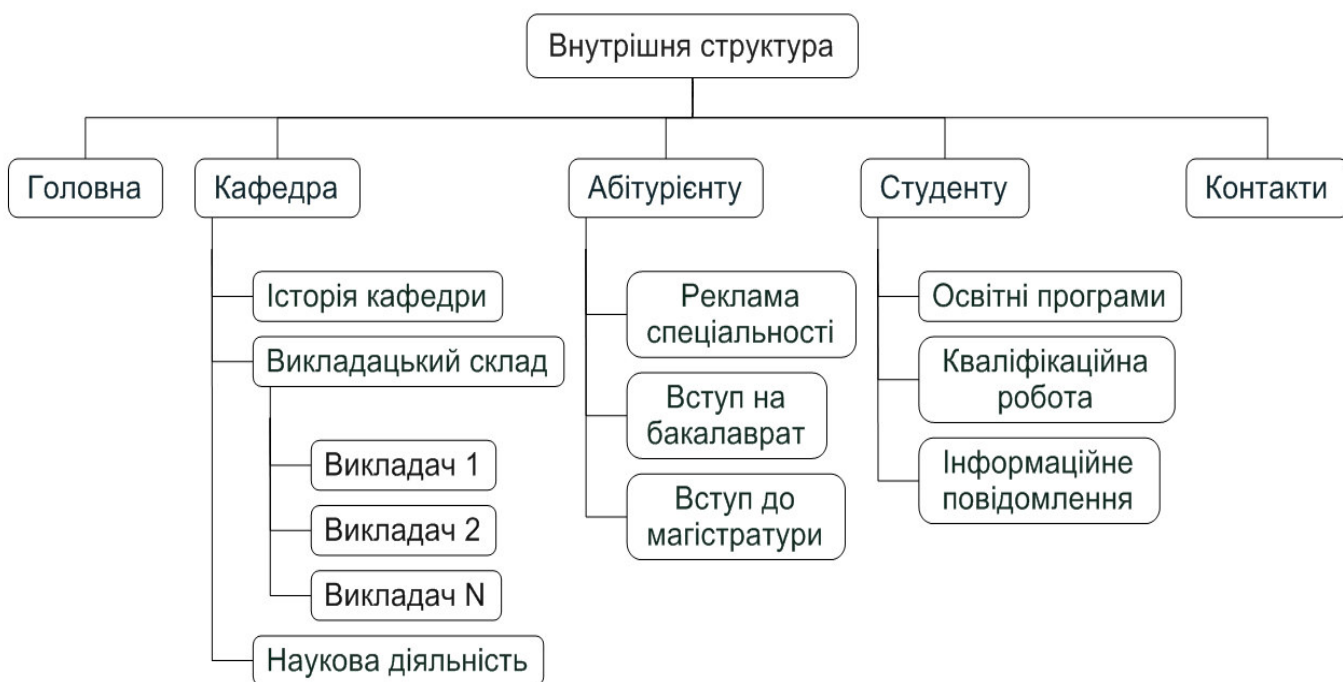


Рис. 3.1. Внутрішня структура веб-ресурсу

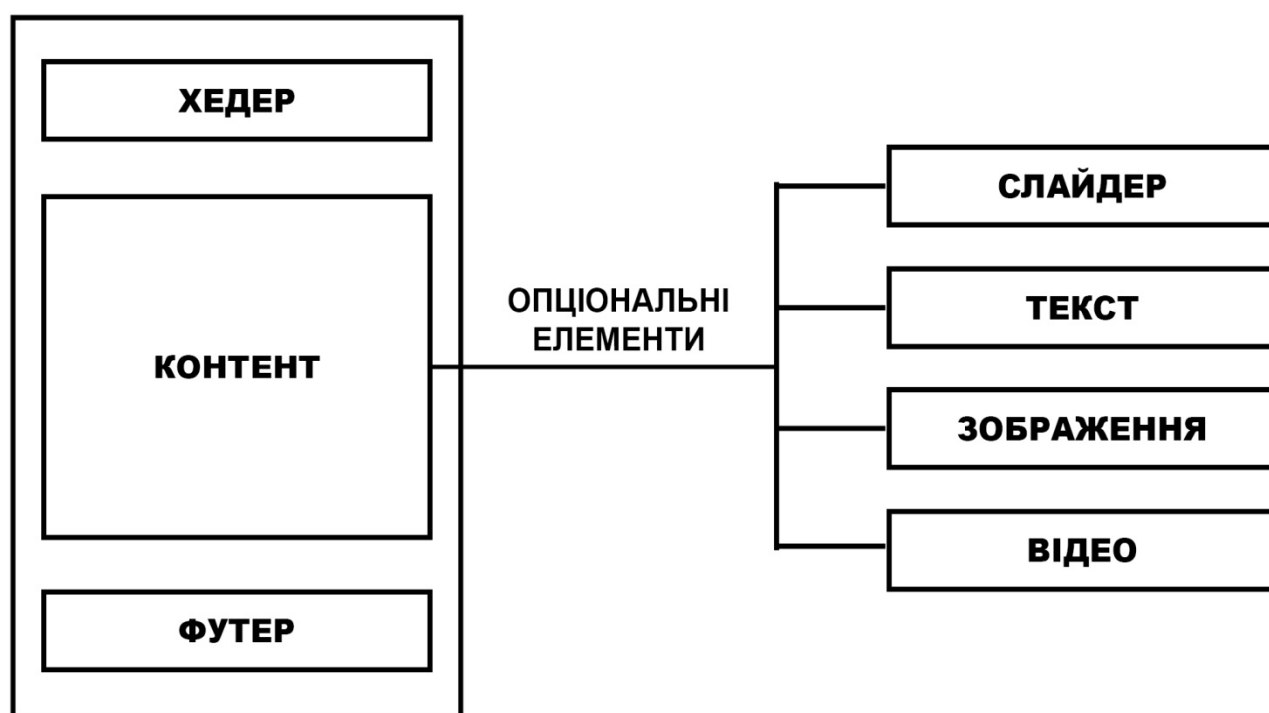


Рис. 3.2. Зовнішня структура веб-ресурсу

Приступаємо до вибору кольорової гами, оскільки вона є дуже важливим фактором впливу на сприйняття інформації. Особливість кольору полягає в тому, що він має здатність впливати на підсвідомому рівні. Розглянувши характерні

властивості різних кольорів, для оформлення веб-ресурсу кафедри було обрано комбінацію трьох кольорів: темно-синього, блакитного та білого.

Темно-синій колір є ключовим в стильовому оформленні, оскільки підкреслює спокій, стабільність та надійність. Він допомагає сконцентруватися на сприйнятті інформації, а отже максимально підходить для оформлення веб-ресурсу структурного підрозділу закладу вищої освіти. Блакитний колір гармонійно поєднується з темно-синім, створюючи при цьому кольорову рівновагу, додаючи відчуття легкості, спокою та сприяє інтенсивній розумовій роботі. Білий колір є нейтральним, а отже органічно поєднується з обраними основними кольорами веб-ресурсу, утворюючи ефект простору та чистоти.

Одним з основних елементів дизайну веб-ресурсу є шрифт тому, що він відповідає за легке сприйняття текстової інформації та повинен гармонійно поєднуватися з іншими складовими.

Виходячи з цього, обрано шрифт *Lato*, який відноситься до групи без засічок (*sans-serif*). Він легко сприймається в основному тексті та заголовках, оскільки легко читається при будь-якому розмірі та надає дизайну веб-ресурсу простоти та мінімалізму.

Створюємо векторний логотип кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова в програмі *CorelDraw*. Дизайнерська ідея логотипу полягає в об'єднанні англійської літери „S”, яка є початковою в слово *surdo* з зображенням рук, які символізують жестову мову. Літера „S” вписується в форму правильного шестикутника, а зображення рук формують зображення літери. Графічне зображення руки малюємо інструментом *Pen*. Потім за допомогою інструменту *Shape Tool* підправляємо її та робимо більш охайною. Далі дублюємо зображення руки, відзеркалюємо в двох напрямках та згідно концепції обидві розміщуємо в шестикутнику. Розроблено два варіанти кольорового виконання: для відображення на світлому та темному фонах (рис. 3.3). Створений логотип відповідає основним правилам дизайну, а також легко сприймається та ідентифікується в будь-якому розмірі.

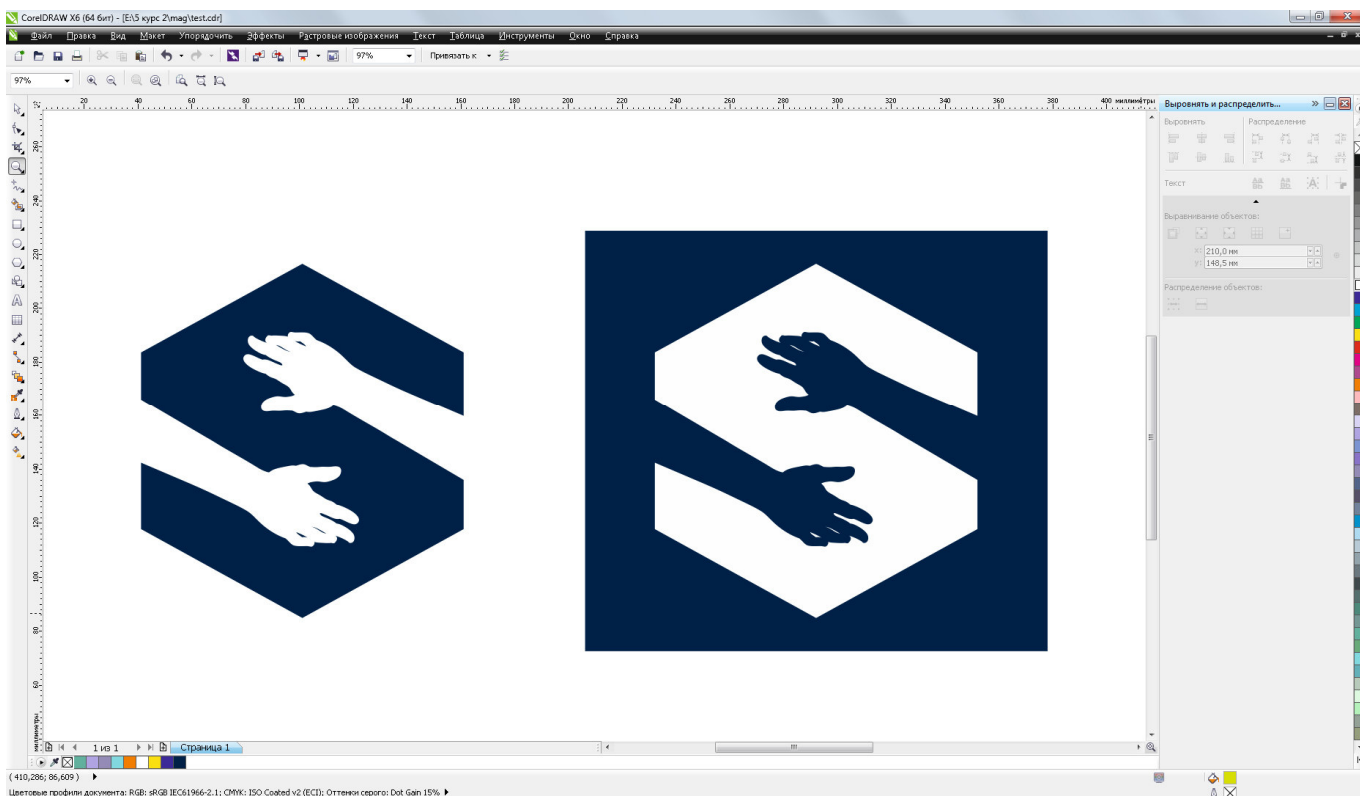


Рис. 3.3. Логотип кафедри

Для розробки дизайну веб-сторінок використовуємо графічний редактор *Corel DRAW*, яким володіє потужним інструментарієм для втілення всіх ідей.

Створюємо дизайн структурних елементів хедера та футера, які повиненні бути наявними на кожній веб-сторінці ресурсу та не змінювати місце розташування (рис. 3.4). Він включає в себе логотип та назву кафедри, панель навігації, пошук. У верхній частині розміщуємо логотип та назву, що розділені між собою декоративною вертикальною білою лінією. В нижній частині розташовуємо панель навігації та піктограму пошуку у вигляді лупи. Панель навігації розроблюємо короткою та лаконічною, щоб не займала багато екранного місця та дозволяла користувачу бачити основні структурні блоки веб-ресурсу. Використовуємо для оформлення хедера темно-синій колір, для тексту – білий, а для активного елементу панелі навігації обираємо блакитний. Футер розташовуємо внизу сторінки, фарбуємо в темно-синій колір та розміщуємо в ньому інформацію про авторські права шрифтом білого кольору. До хедера та футера застосовуємо обраний шрифт *Lato* з ранжуванням розміру у відповідності до значущості елементів.

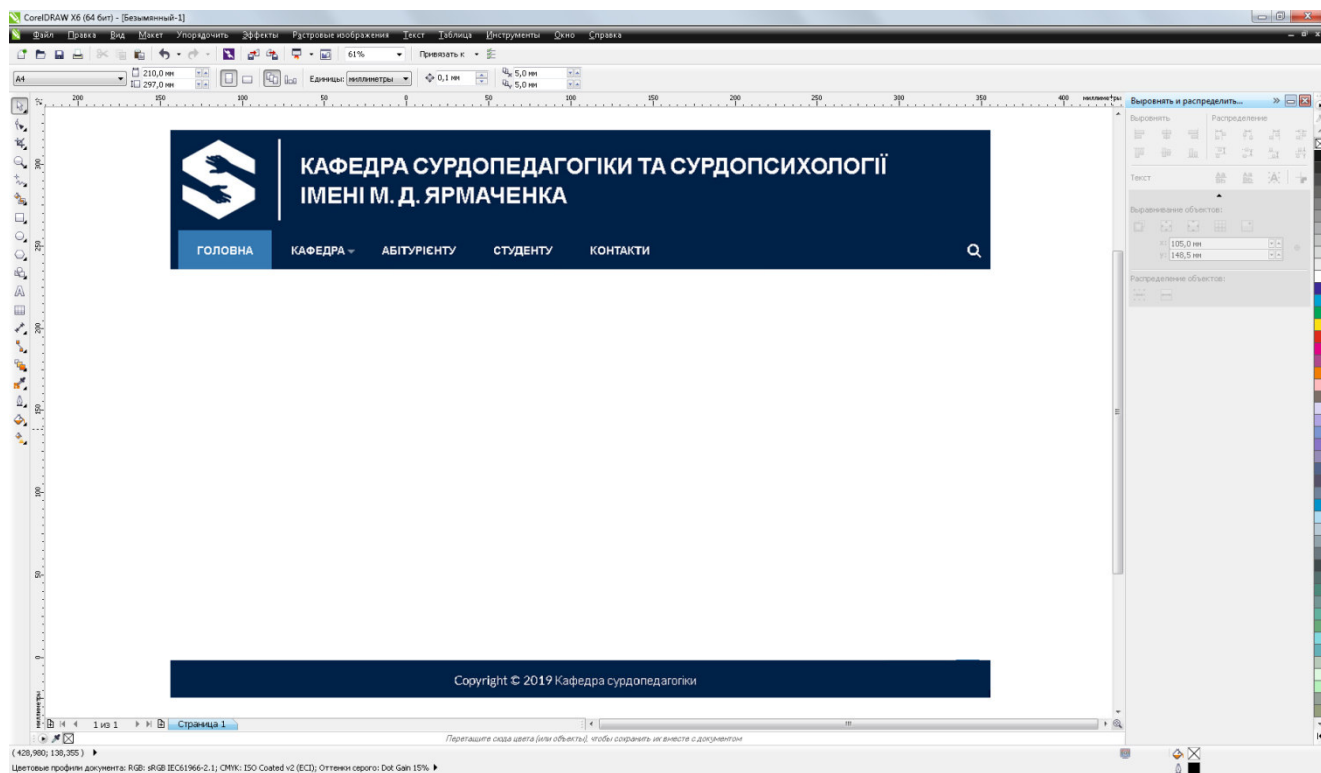


Рис. 3.4. Зовнішній вигляд хедера та футера

Для головної сторінки проектуємо слайдер, який дозволяє користувачам побачити анонс актуальної інформації з можливістю більш детального ознайомлення з нею (рис. 3.5). Він складається з прямокутного блоку для ілюстративного матеріалу, в якому по центру розташовано заголовок та кнопку переходу на сторінку з повним викладенням інформації. Додаємо органи керування: зліва та справа у вигляді квадратів з вказівниками у вигляді стрілок, знизу у вигляді кружечків, що вказують кількість слайдів та активну сторінку слайдера.

Формуємо дизайн блоку новин для головної сторінки, що має форму прямокутника залитого білим кольором, який змінюється на блакитний при наведенні курсору. Зверху блоку розташовуємо заголовок шрифтом *Lito* розміром 30 *px*. Під ним розміщуємо зліва фотографію, а зліва певний фрагмент тексту новини розміром 16 *px*. З самого низу створюємо інформаційно-навігаційну секцію у вигляді рядка, з використанням акцентуючих елементів у вигляді декоративних піктограм, в якому знаходяться: автор матеріалу, дата публікації, рубрика та кнопка для переходу на повну версію новини. Для кращої орієнтації на сторінці серед новин, відокремлюємо кожну з них знизу декоративною синьою лінією.

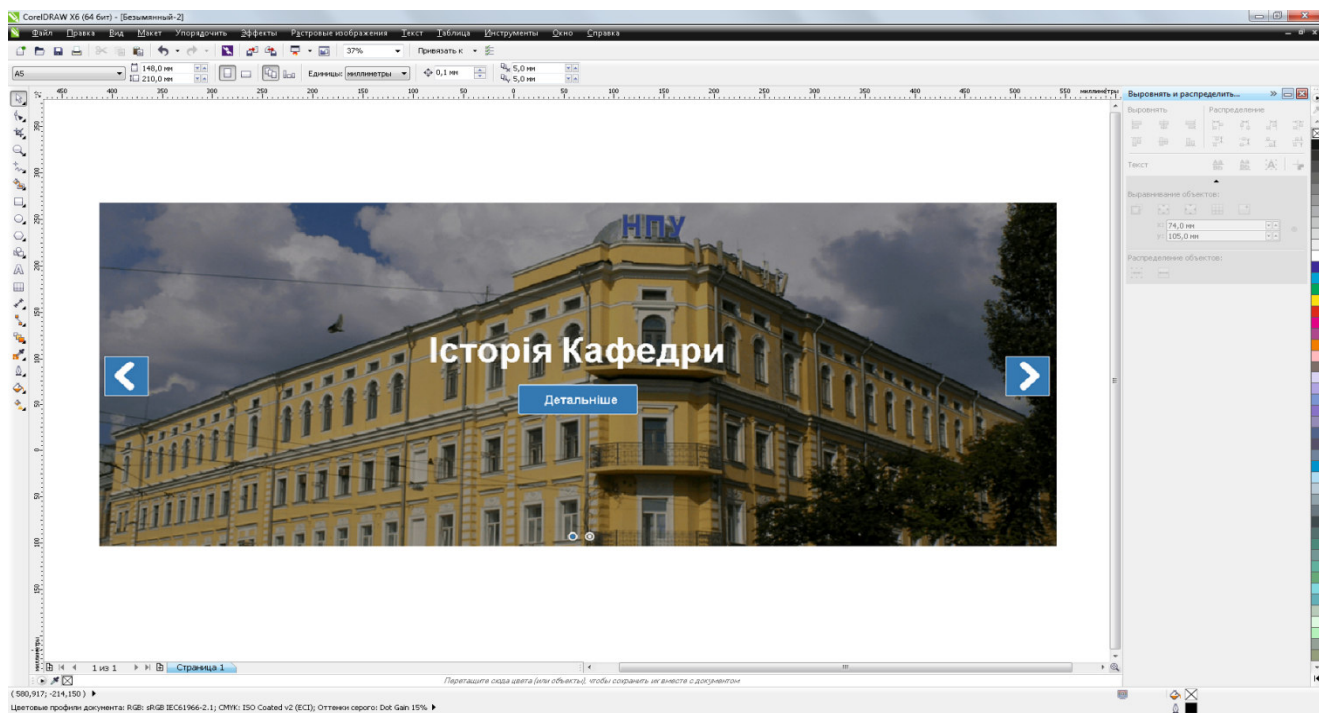


Рис. 3.5. Дизайн слайдера

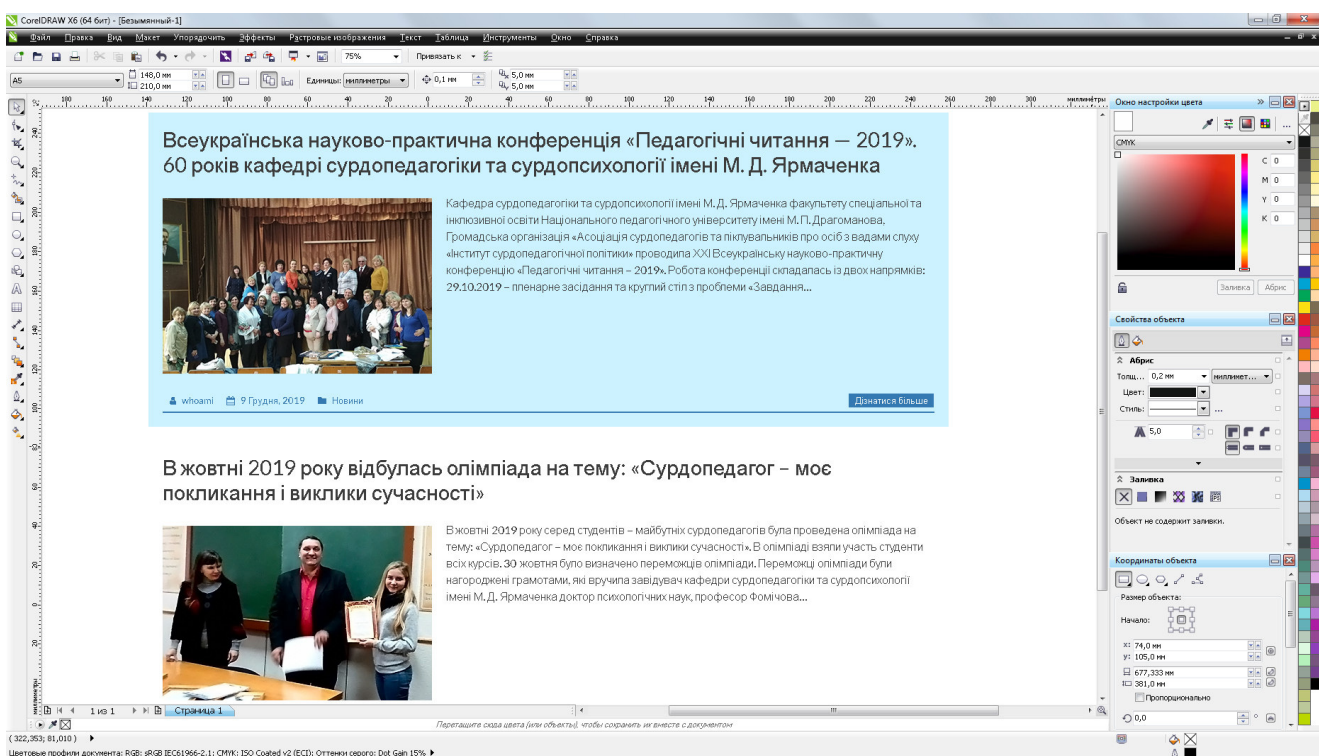


Рис. 3.6. Дизайн блоку новин

Створюємо дизайн сторінок структурних розділів, який розбивається на дві зони. В верхній частині розташовуємо заголовок та короткий опис. В нижній частині розміщуємо блоки переходу до підрозділів, які включають фото та назву.

Розроблюємо дизайн при наведенні на блок із збільшенням розміру зображення, накладення на нього синього коліру та піктограмою для переходу до обраного розділу (рис. 3.7). Цей дизайн використовуємо для всіх подібних сторінок.

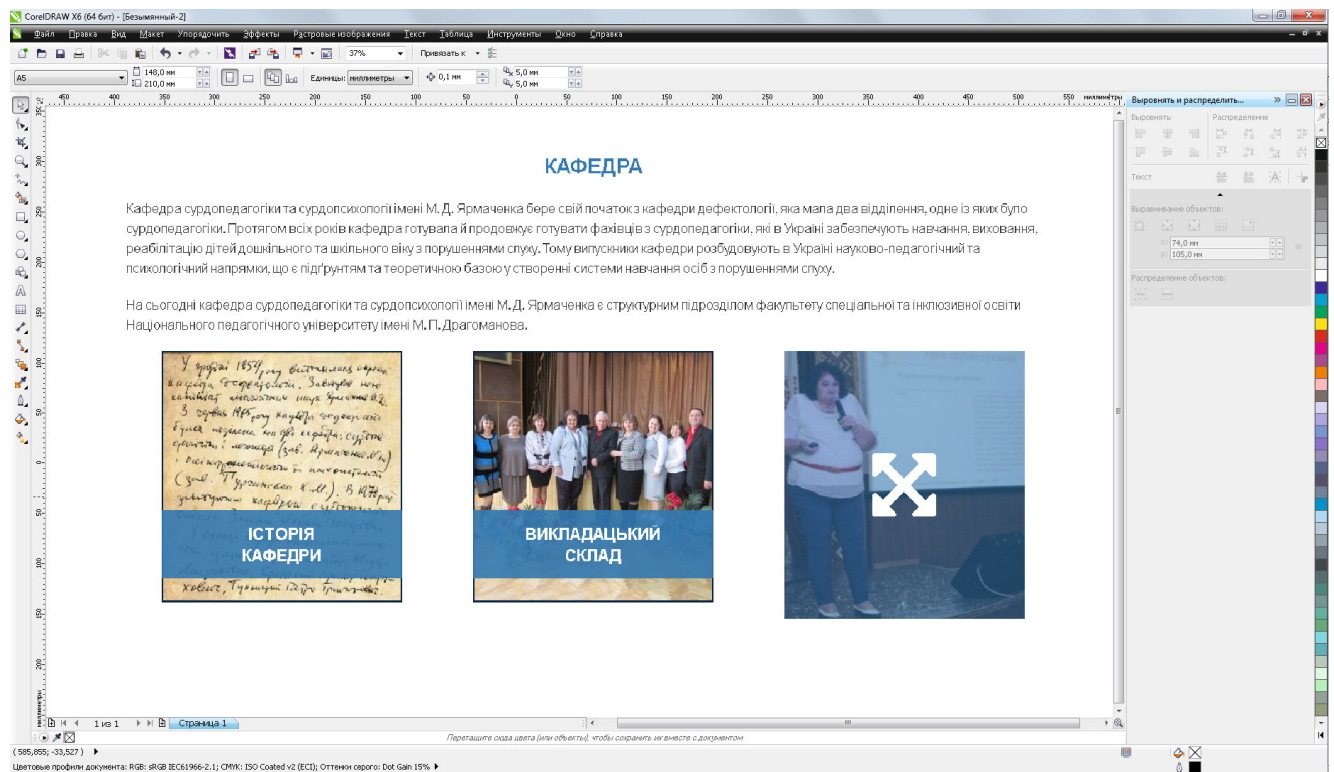


Рис. 3.7. Дизайн розділу

Проектуємо зовнішній вигляд сторінки викладацького складу (рис. 3.8). Розташовуємо фотографії викладачів у вигляді блоків прямокутної форми, знизу яких розміщуємо напис прізвища та ініціалів шрифтом *Lato* білого кольору на синьому фоні, що має десять відсотків прозорості. При наведенні курсору передбачаємо збільшення розміру фотографії та появу затемнення на якому розташовуємо піктограму для переходу на сторінку обраного викладача. Для того, щоб блоки не зливалися з фоном сторінки застосовуємо темно-синю суцільну рамку товщиною два пікселя.

Створюємо сторінку викладача, яку розділяємо на дві секції: ілюстративну та змістовну. У змістовній секції передбачаємо рубрики (посада та наукові ступені, основні наукові праці, профілюючі дисципліни, посилання на *Google Scholar*), які відокремлені одна від одної кольором заливки та відступами (рис. 3.9).

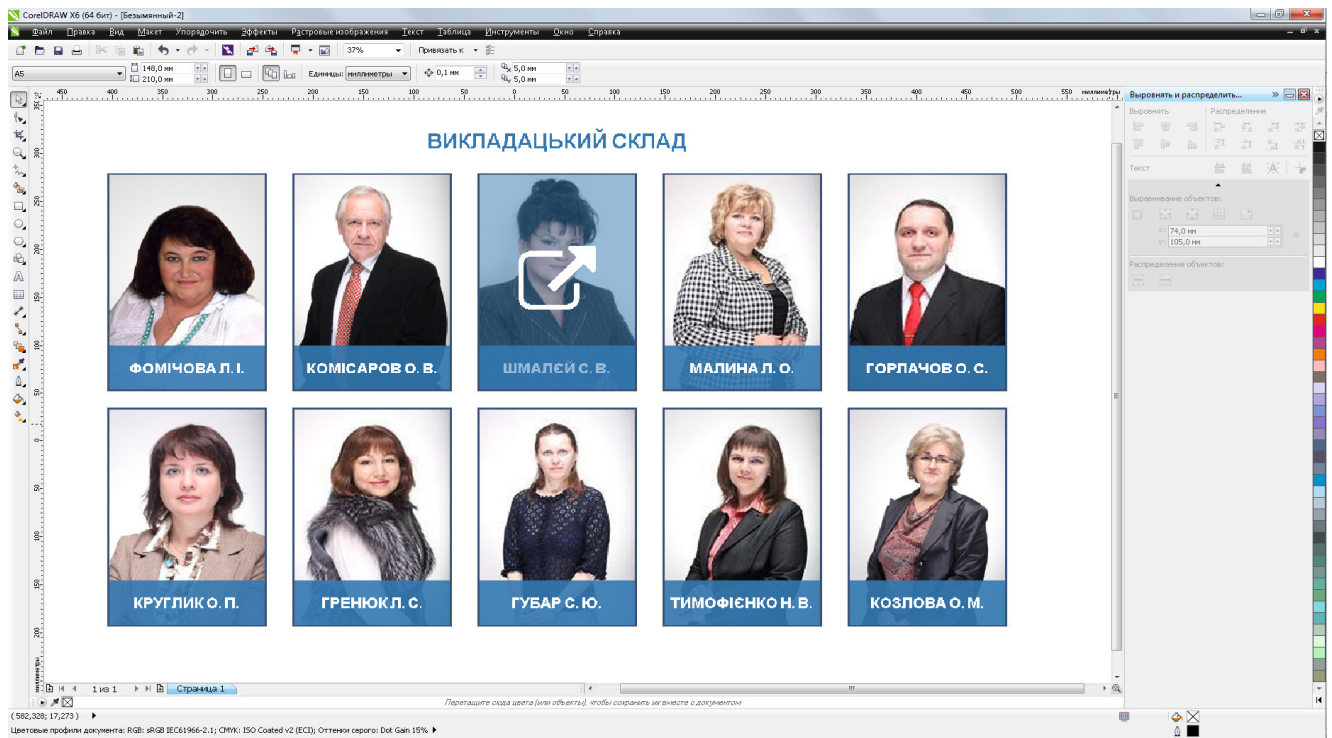


Рис. 3.8. Дизайн сторінки викладацького складу

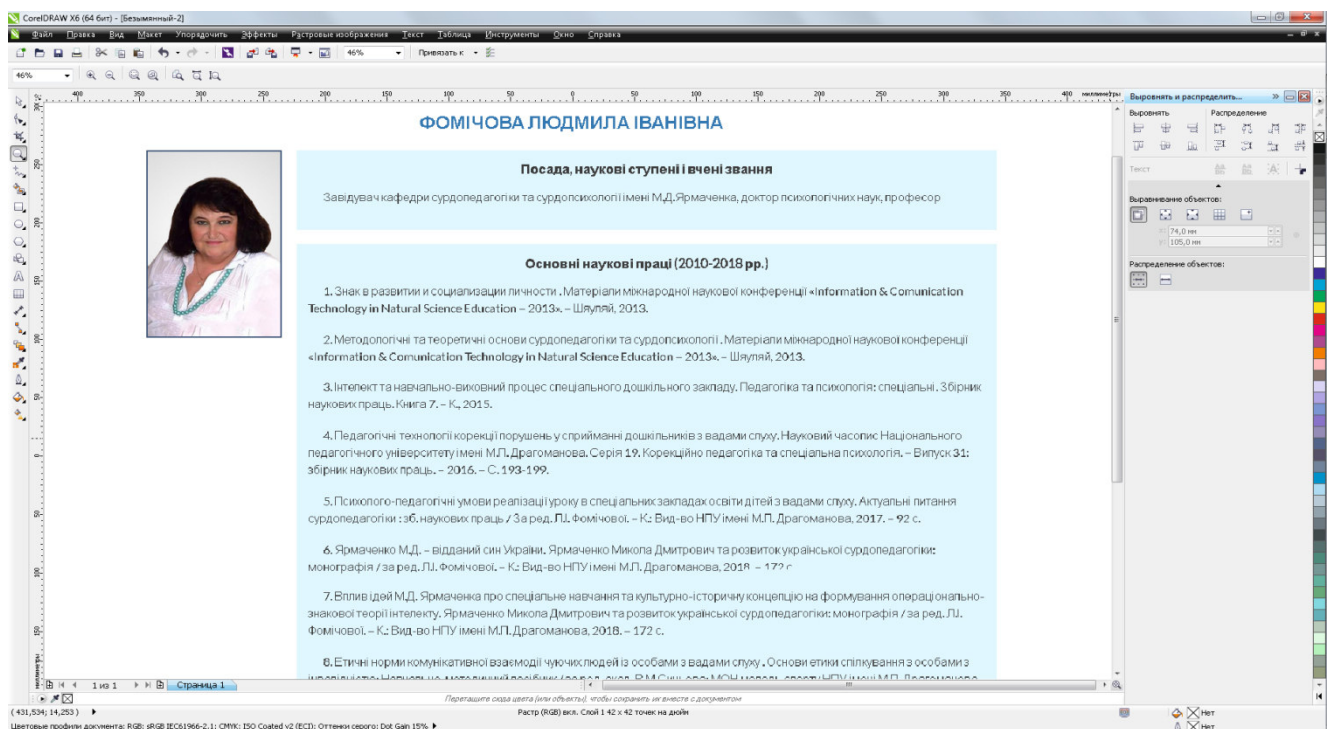


Рис. 3.9. Дизайн сторінки викладача

Формуємо три стани відображення навігації по структурним підрозділам, які відображають розташування на першій, останній та проміжній сторінках. Також передбачаємо адаптивний варіант (рис. 3.10).

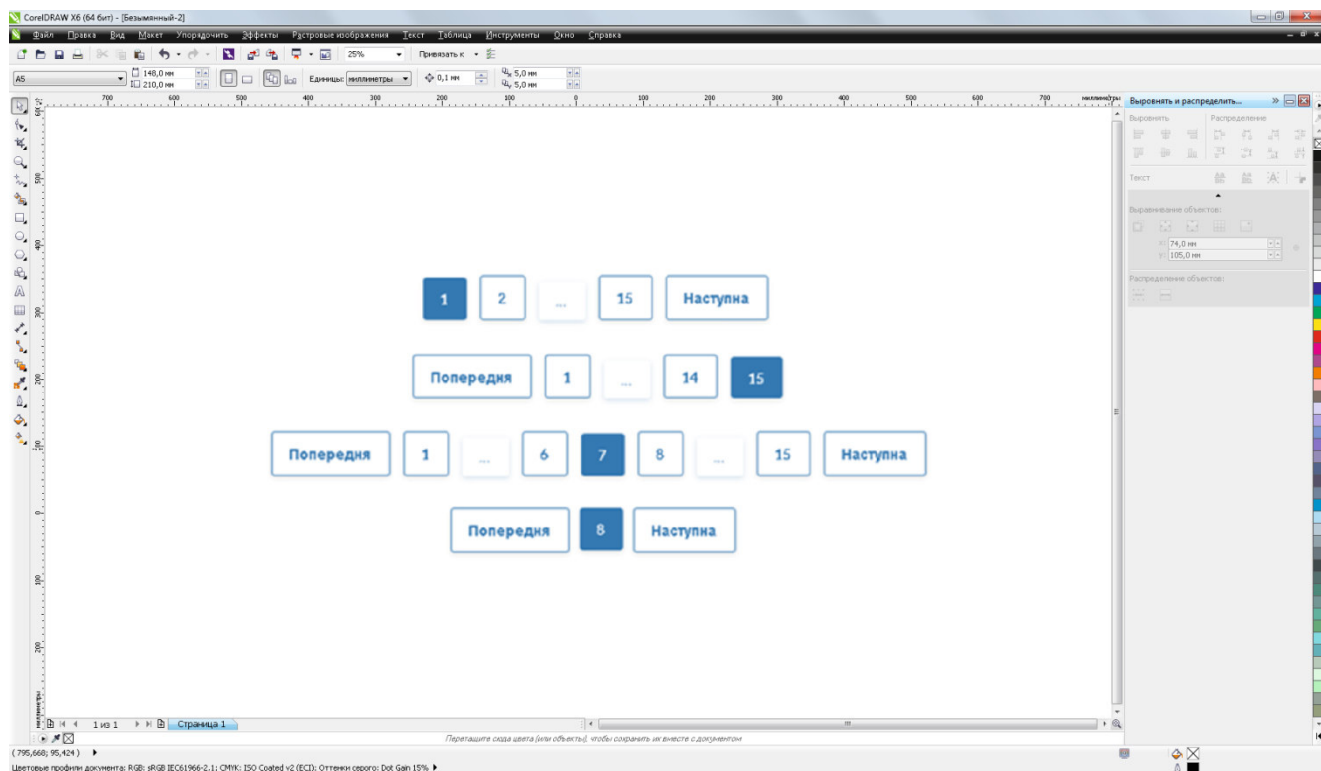


Рис. 3.10. Стани панелі навігації

На фінальному кроці розробки дизайнерського рішення розроблюємо варіанти адаптивності для кожної з сторінок, щоб забезпечити коректне відображення інформації на екранах будь-якого розміру.

3.2. Написання програмного коду та верстка веб-ресурсу

Обираємо хостинг, на який інсталуємо *CMS WordPress*, та бронюємо доменне ім'я. Підключаємо базу даних для розміщення та зберігання вмісту веб-ресурсу. Вивчаємо наявні теми з доступної колекції, що найбільш подібні до розробленої концепції дизайну, серед яких робимо вибір на користь *AccessPress Basic*.

Після цього на її основі створюємо дочірню тему „*Surdo Theme*”, яка розширює функціонал базової теми, дозволяючи при цьому змінювати та доповнювати можливості батьківської теми. Створюємо для цього нову папку на хостингу та додаємо до неї файл стилів з підключенням обраної теми та активуємо її на ресурсі. Імпортуємо до теми зображення для надання більшої інформативності та привабливості (рис. 3.11).

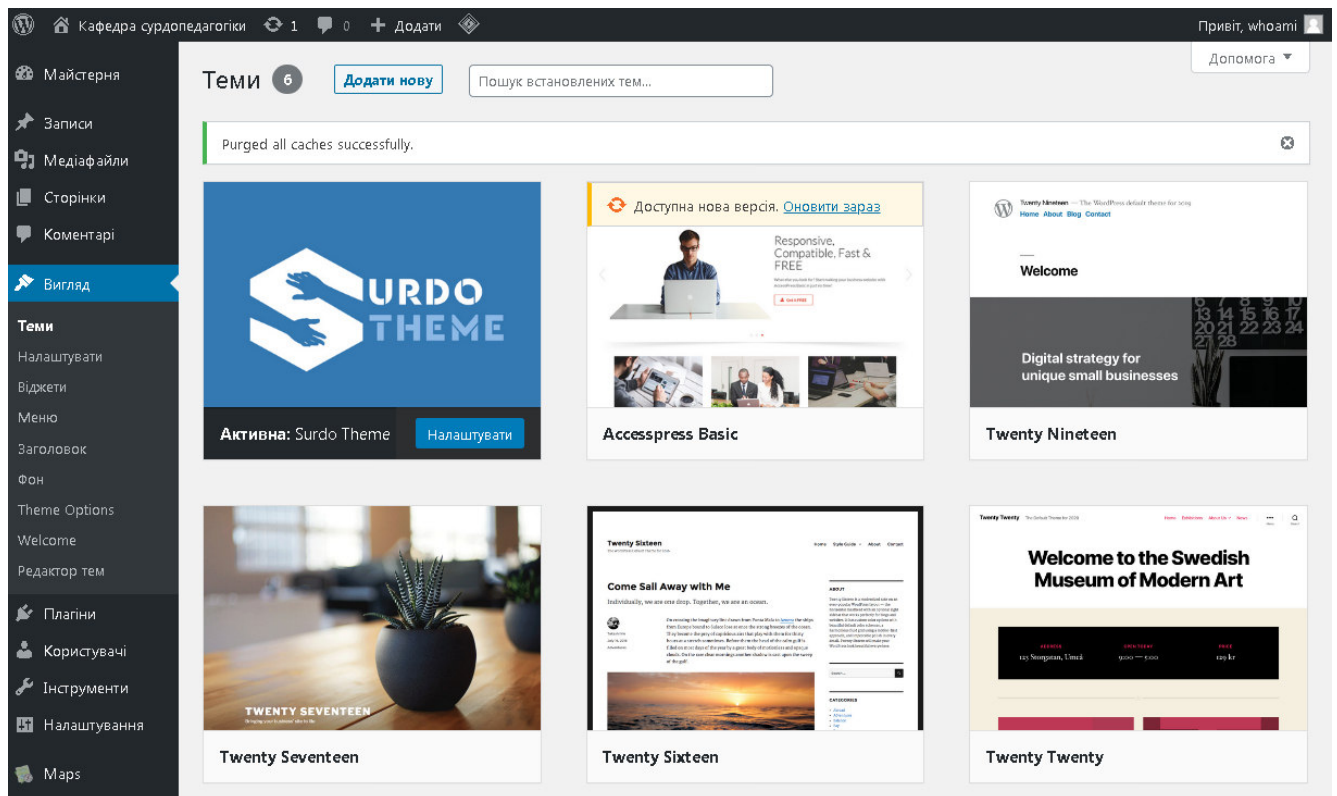


Рис. 3.11. Дочірна тема

Починаємо переносити створений сайт до *CMS WordPress*. В першу чергу завантажуюємо всі необхідні плагіни: *Classic Editor*, *Yoast SEO*, *TinyMCE Advanced*, *Smush*, *RoboGallery* та *WP Google Maps* (рис. 3.12).

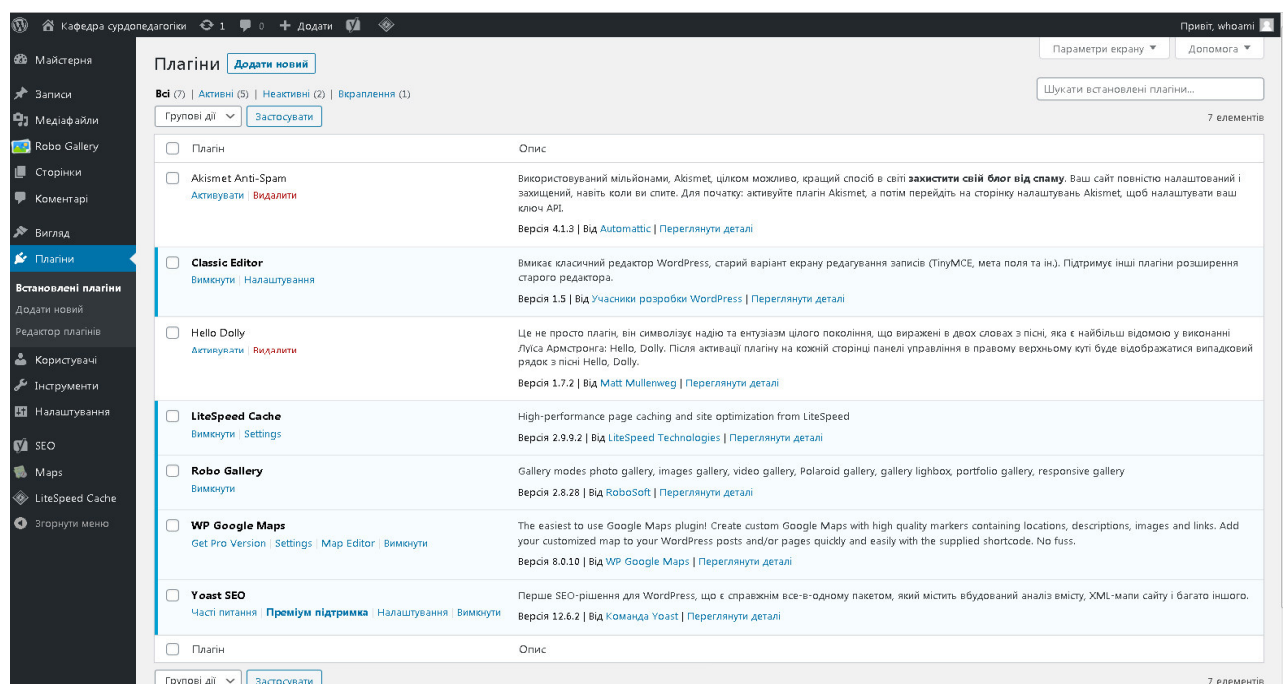


Рис. 3.12. Інсталяція плагінів

Весь наданий ілюстративний матеріал оброблюємо за допомогою програми *Adobe Photoshop*, а саме редагуємо розміри та дозволи, виконуємо корекцію, вирізаємо необхідні фрагменти та завантажуюмо їх в медіа бібліотеку (рис. 3.13).

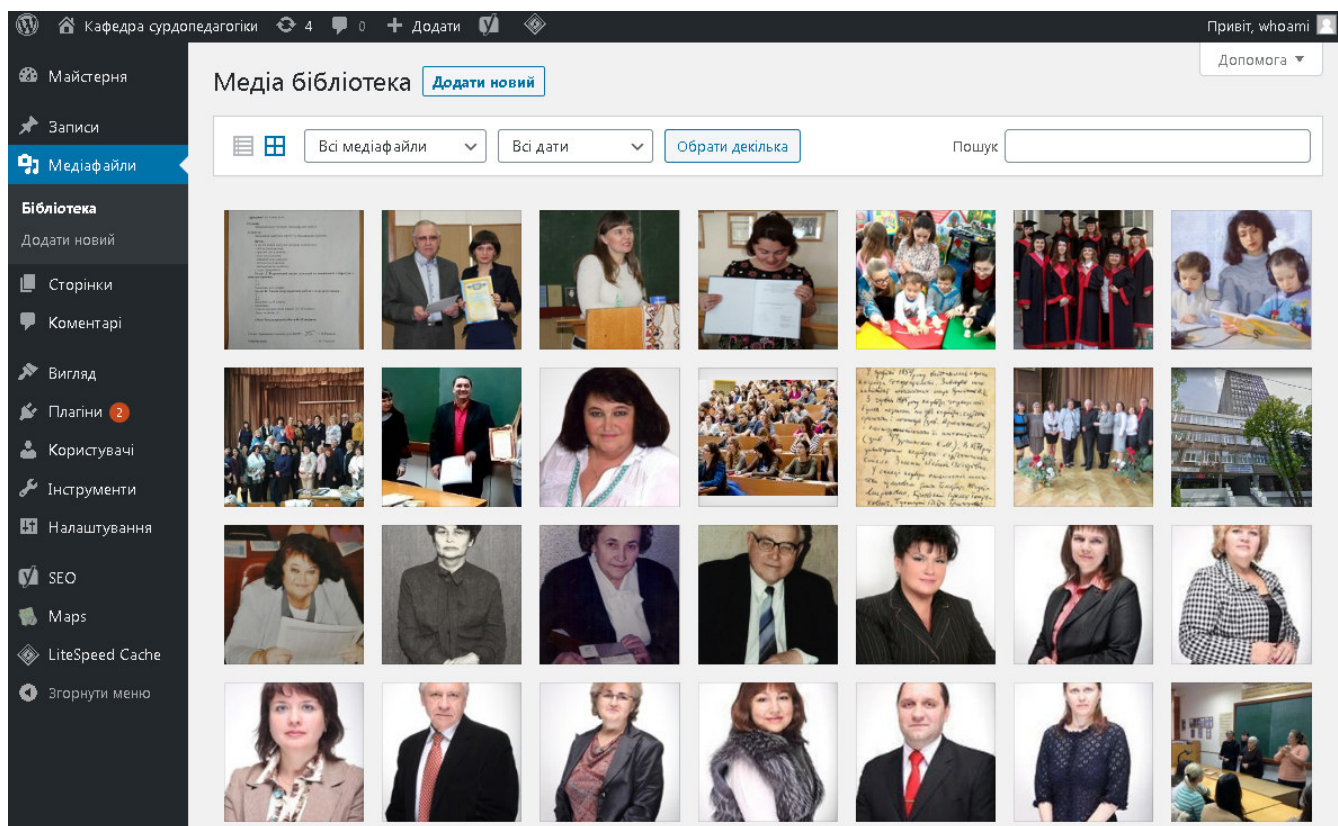


Рис. 3.13. Медіа бібліотека

Розташовуємо весь опрацьований текстовий та ілюстраційний матеріал у файлове сховище. Формуємо за допомогою візуального редактору меню у відповідності з розробленою внутрішньою структурою веб-ресурсу (рис. 3.14).

Модифікуємо хедер шляхом доповнення *HTML*-коду для додавання зображення логотипу та назви кафедри. Для зручності навігації прописуємо гіперпосилання для логотипу на головну сторінку. За допомогою *CSS* описуємо стилі для оформлення хедера, а саме задаємо позиціонування логотипу, накреслення та розмір шрифту, колір тексту та фону, додаємо відокремлюючу вертикальну білу лінію. Прописуємо ефекти при наведенні курсору та виділення активних елементів, а також позиціонування випадаючого списку меню (рис. 3.15). Створюємо в *CSS* можливість адаптації для хедера за допомогою *@media*, прописуючи зміни розташування елементів для необхідних розмірів екранів.

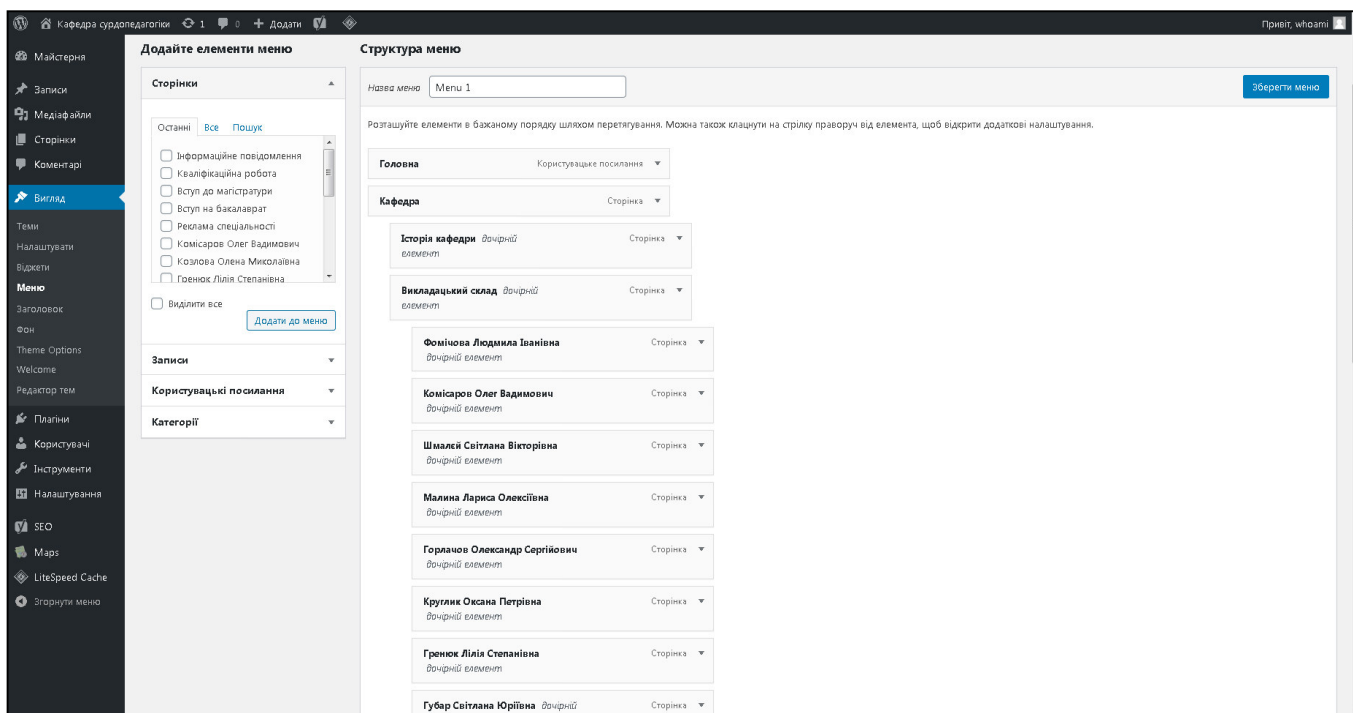


Рис. 3.14. Налаштування структури меню

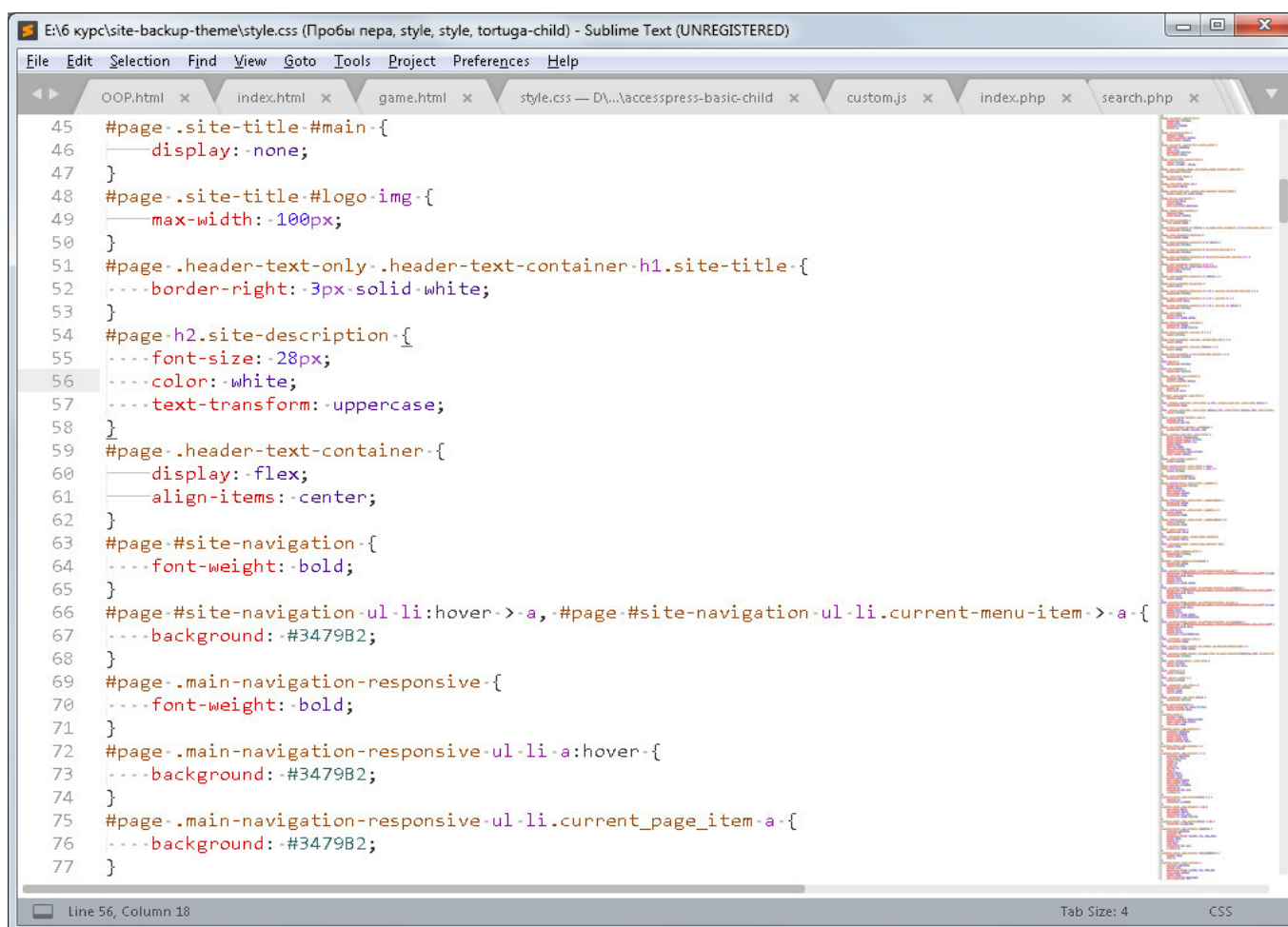


Рис. 3.15. Верстка хедера

Імпортуємо на головну сторінку *bxSlider*, який написано за допомогою бібліотеки *jQuery*. Виконуємо його налаштування, редагуючи параметри під вимоги веб-ресурсу (рис. 3.16). Використовуємо *CSS* для заміни органів керування та кольорової гами у відповідності до дизайнерського рішення.

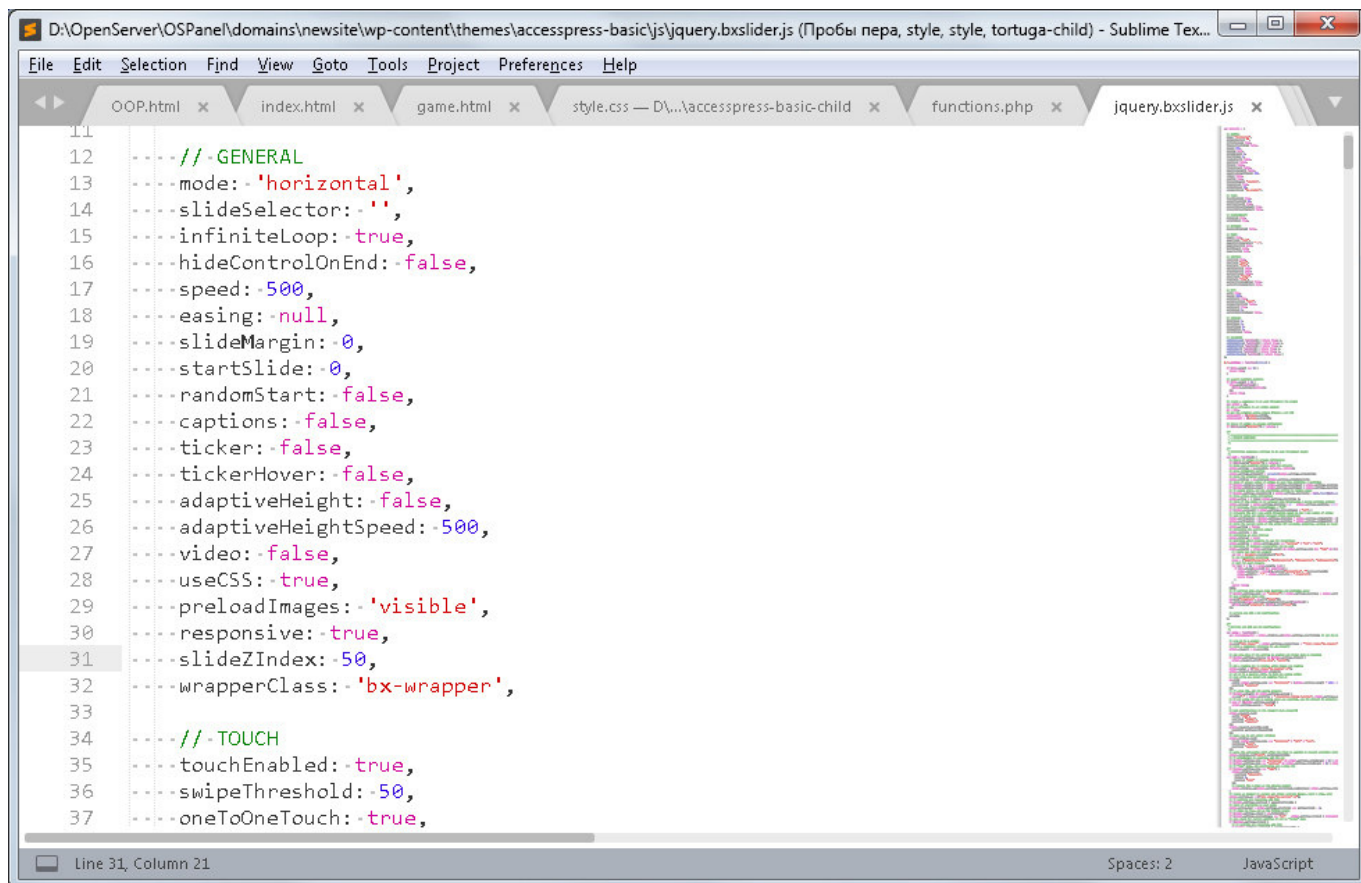


Рис. 3.16. Налаштування слайдеру

Встановлюємо на головній сторінці відображення новин у кількості чотирьох постів. Доповнюємо дочірню тему *RHP*-кодом (рис. 3.17), який додає в документ елементи керування переходу між сторінками новин (додаток А). У коді прописуємо відображення першої та останньої сторінок, а також опціональні кнопки переходу на наступну або попередню. При збільшенні кількості постів програмуємо появу трьох кнопок з вказанням активної сторінки, а по бокам декоративні елементи, що вказують на наявність інших сторінок, нумерація яких не відображається на панелі керування. За допомогою *CSS* задаємо стилі (рис. 3.18), а саме форму кнопок та їх дизайн, виділення для активного елементу керування та *hover*-ефект при наведенні курсору. Також з використанням *@media* вказуємо параметри адаптивності.

```
D:\OpenServer\OSPanel\domains\newsite\wp-content\themes\accesspress-basic-child\index.php • (Пробы непа, style, style, tortuga-child) - Sublime Text (...)
```

```
File Edit Selection Find View Goto Tools Project Preferences Help
```

```
OOP.html x index.html x game.html x style.css — D:\...accesspress-basic-child x index.php x functions.php x
```

```
90 <?php
91 if(!function_exists('surdo_numeric_posts_nav'))-{
92     function surdo_numeric_posts_nav(){
93         .....
94         $prev_arrow=-is_rtl()?-'Previous':-'Next';
95         $next_arrow=-is_rtl()?-'Next':-'Previous';
96         .....
97         global $wp_query;
98         $total=-$wp_query->max_num_pages;
99         $big=-999999999;
100         if($total>1){
101             if(!$current_page=-get_query_var('paged'))
102                 $current_page=1;
103             if(get_option('permalink_structure'))-{
104                 $format='page/%#%/';
105             }-else-{
106                 $format='&paged=%#%';
107             }
108             echo wp_kses_post(paginate_links(array(
109                 'base'=>str_replace($big,'%#%',esc_url(get_pagenum_link(
110                     $big))),
111                 'format'=>$format,
112                 'current'=>max(1,get_query_var('paged')),
113                 'total'=>$total,
114                 'mid_size'=>1,
115                 'type'=>'list',
116                 'prev_text'=>'Попередня',
117                 'next_text'=>'Наступна',
118             )));
119         }
120     }
121 }
122 <?php
123 .....echo '<div class="text-center">';
124 .....surdo_numeric_posts_nav();
125 .....echo '</div>';
126 ?>
```

```
Line 99, Column 26 Spaces: 4 PHP
```

Рис. 3.17. PHP-код елементів керування



Рис. 3.18. Зовнішній вигляд елементів керування

Використовуємо стандартні функції бібліотеки jQuery для створення вікна пошуку по ресурсу, яке з'являється при натисканні на піктограмі. Модальне вікно включає в себе рядок пошуку та кнопку його активації (рис. 3.19). При його відкритті прописуємо затемнення основного екрану та у верхньому правому куті програмуємо кнопку закриття.

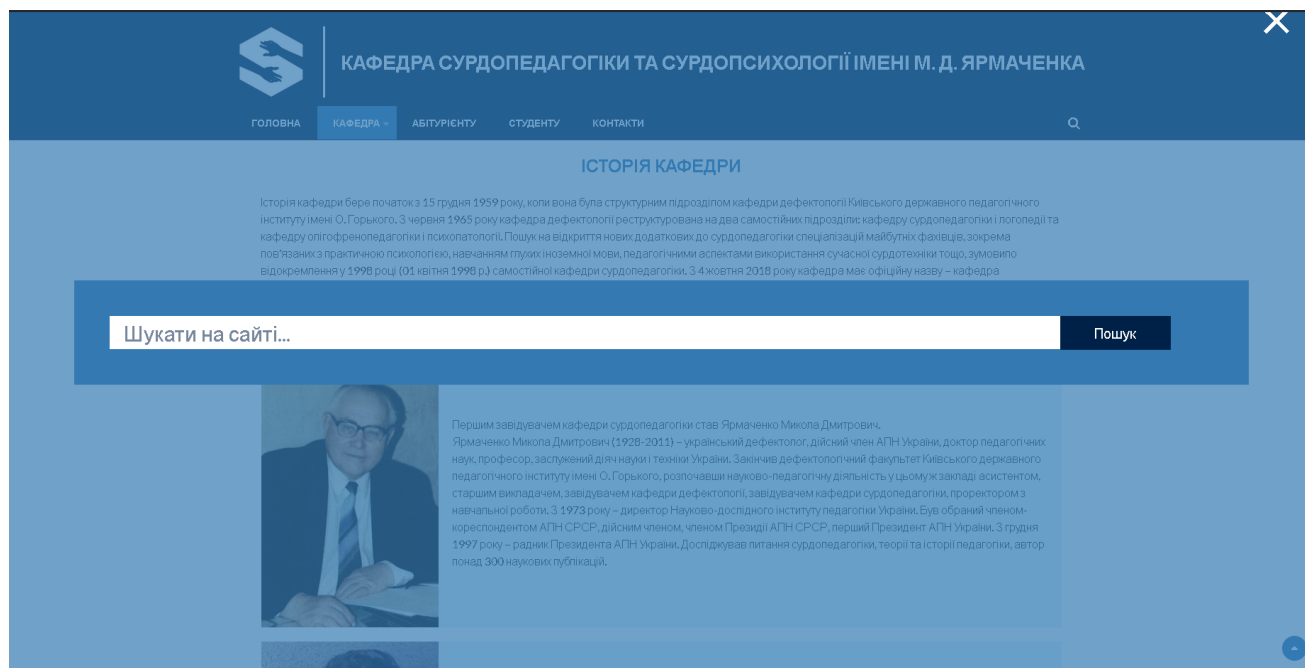


Рис. 3.19. Вікно пошуку

Виконуємо налаштування для мапи в плагіні *WP Google Maps*: обираємо робочу платформу, вказуємо ширину та висоту, задаємо тип та масштаб відображення, редагуємо зовнішній вигляд маркера, підключаємо необхідні органи керування (рис. 3.20).

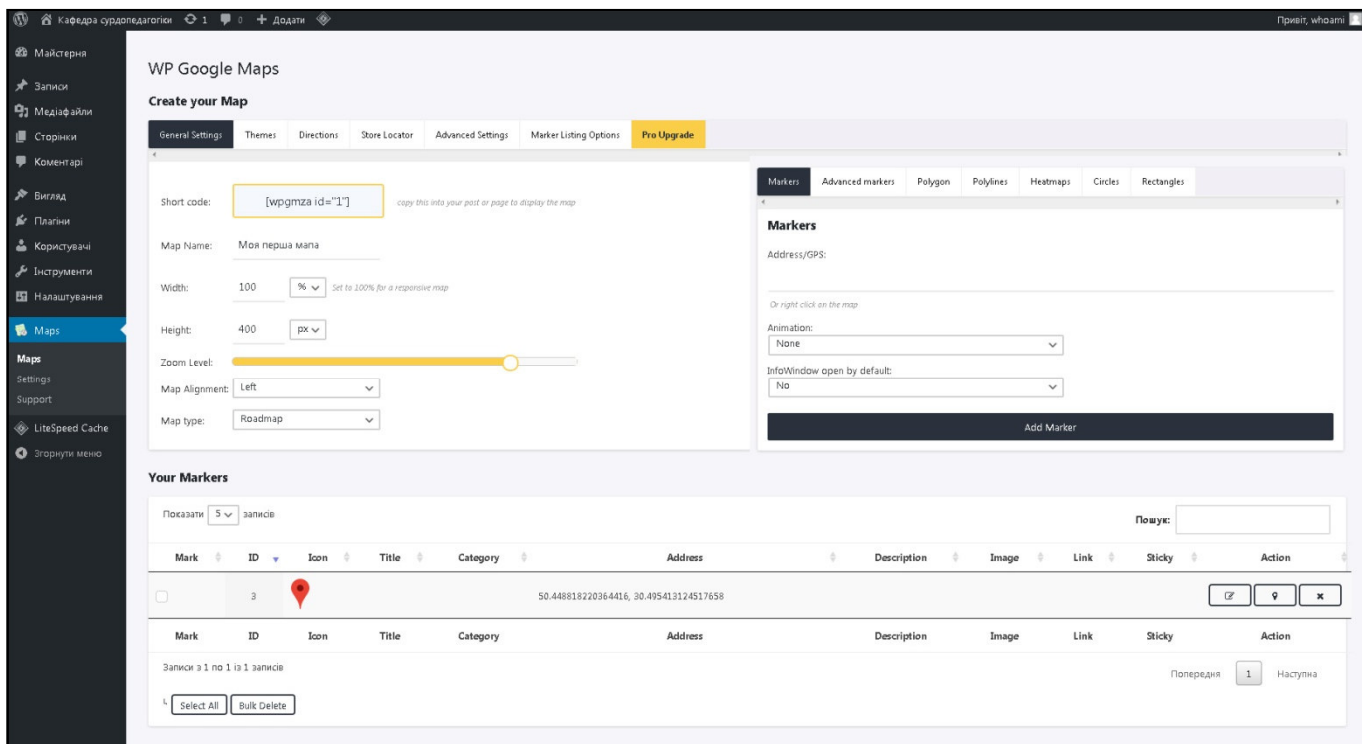


Рис. 3.20. Налаштування *WP Google Maps*

Для сторінки контакти додаємо зображення гуманітарного корпусу в якому розташовано кафедру, контактну інформацію (*e-mail*, телефон та адресу) та шорткод створеної мапи (рис. 3.21). Описуємо за допомогою *CSS* зовнішній вигляд та розташування елементів. Контактну інформацію розміщуємо на блакитному фоні з використанням піктограм, що візуалізують подану інформацію.

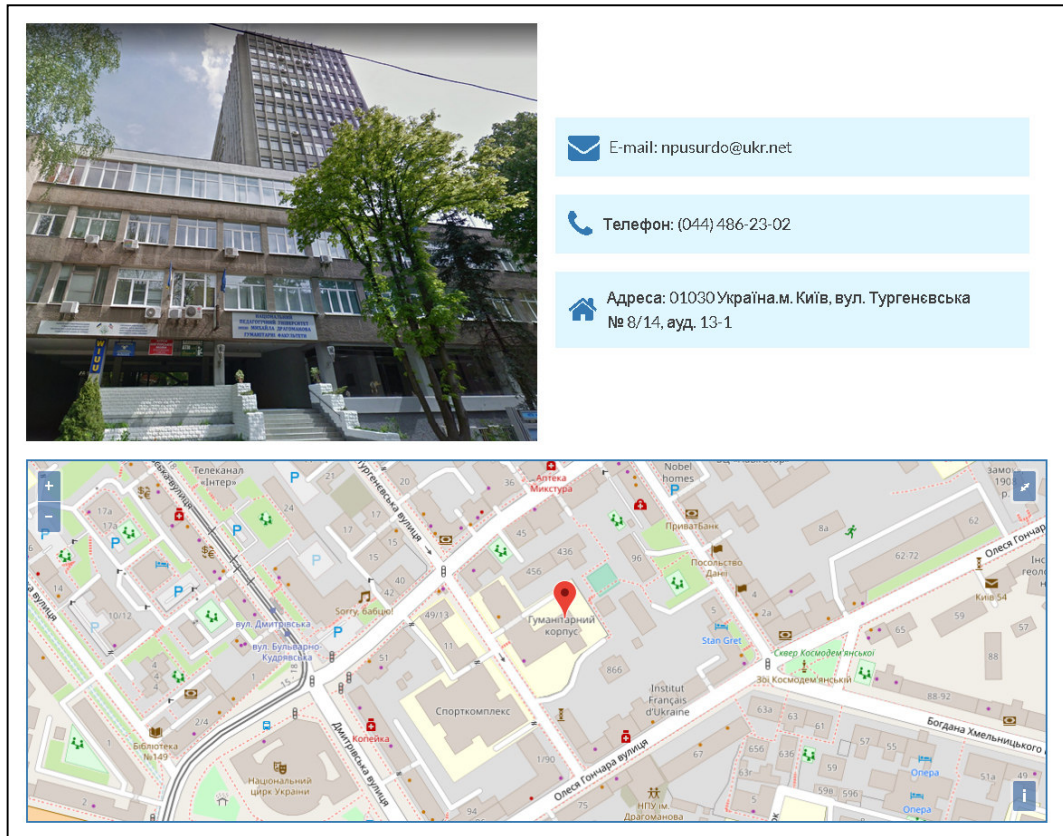


Рис. 3.21. Зовнішній вигляд сторінки контакти

Верстаємо код для відображення дизайну сторінок структурних розділів. Для блоків вибору застосовуємо технологію розміщення *Flexbox*. За допомогою псевдо-елементів розміщуємо компоненти та приховуємо їх за замовчуванням. Задаємо параметри переходу для всіх властивостей та час виконання півсекунди. Блок анонсу підрозділу формуємо у вигляді прямокутника та розміщуємо в ньому ілюстративний контент та назву прописними літерами. Вказуємо послідовність дій при наведенні курсору, а саме зміну розміру та проявлення синього прозорого фону з появою піктограми для переходу до повної інформації. При цьому приховуємо назву підрозділу та збільшуємо масштаб ілюстративного вмісту блоку (рис. 3.22).

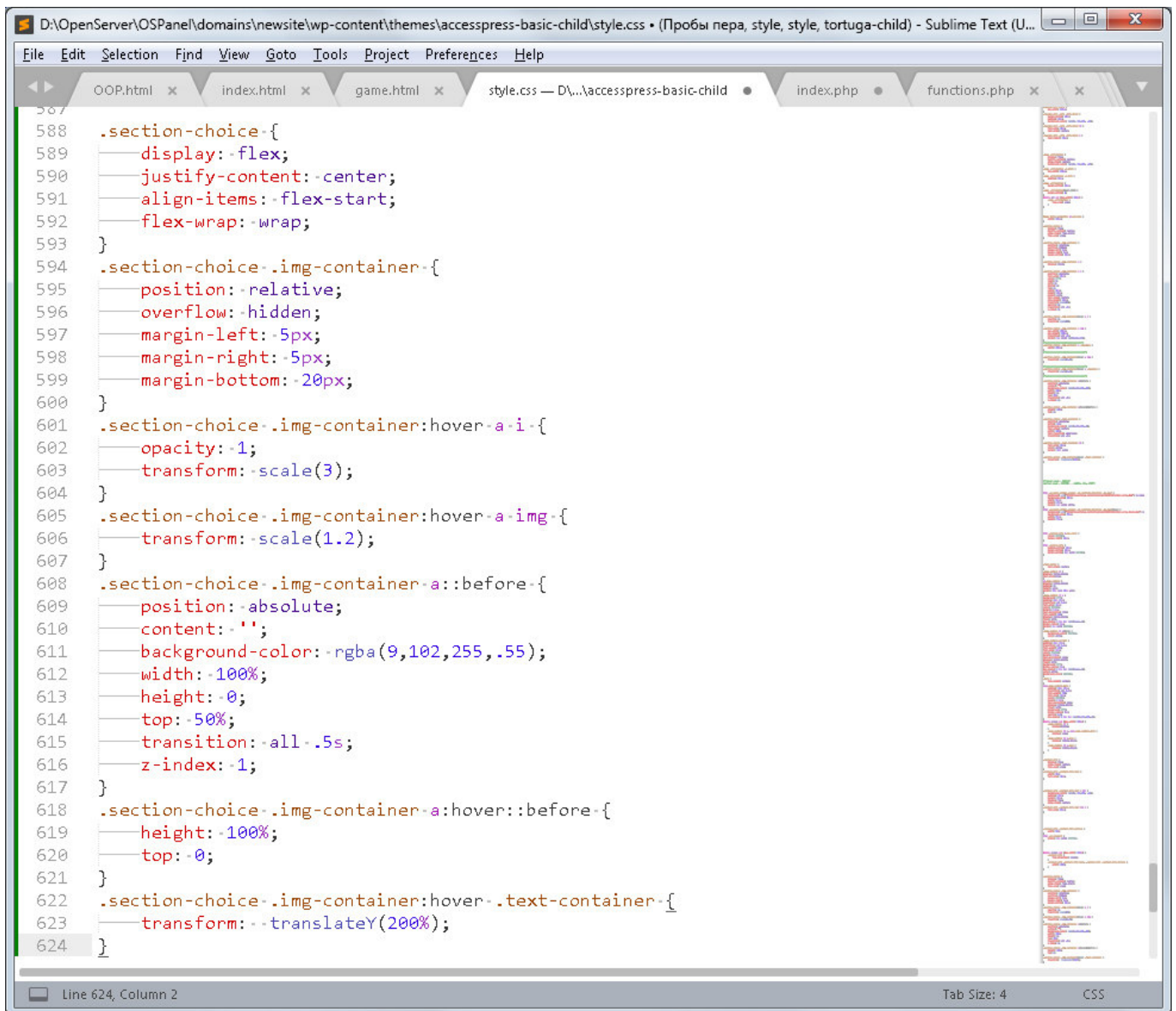


Рис. 3.22. CSS-код реалізації структурного розділу

Створюємо сторінку освітніх програм, що включає в себе заголовок, фотографію та функціональний блок, який розділено у відповідності до навчального плану на чотири окремі напрямки програм (бакалавр та три варіанти магістратури відповідно до термінів навчання). Кожна група освітніх програм містить список *pdf*-посилань, які розкриваються знизу від заголовку блоку за допомогою кліка (рис. 3.23). Реалізуємо дану анімацію з використанням можливостей бібліотеки *jQuery* та стилізуємо зовнішній вигляд з прописуванням ефектів при наведенні за допомогою мови *CSS*. Додаємо можливість переходу посилання на нову вкладу в браузері з відкриттям обраного файлу (додаток Б).

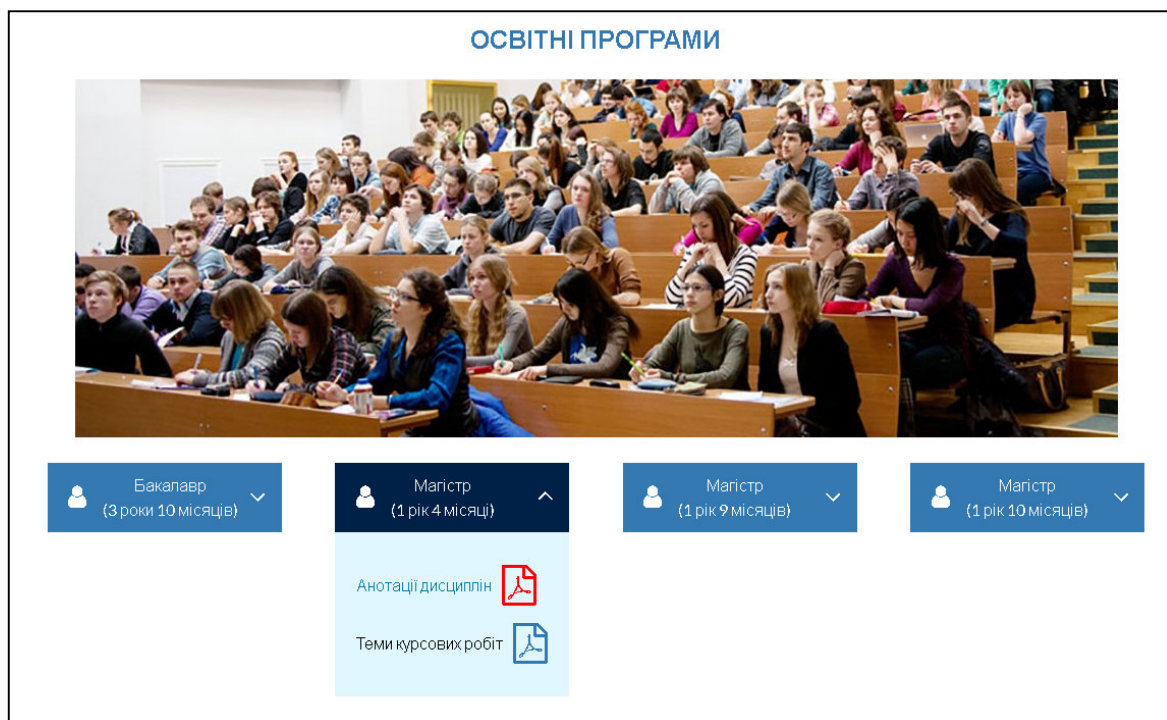


Рис. 3.23. Сторінка освітніх програм

Виконуємо налаштування *SEO*-оптимізації за допомогою плагіна *Yoast SEO*, а саме вказуємо правильне відображення *URL*-адрес на сторінках та постах, таксономію рубрик та форматів, „хлібні крихти” та *RSS* (рис. 3.24).

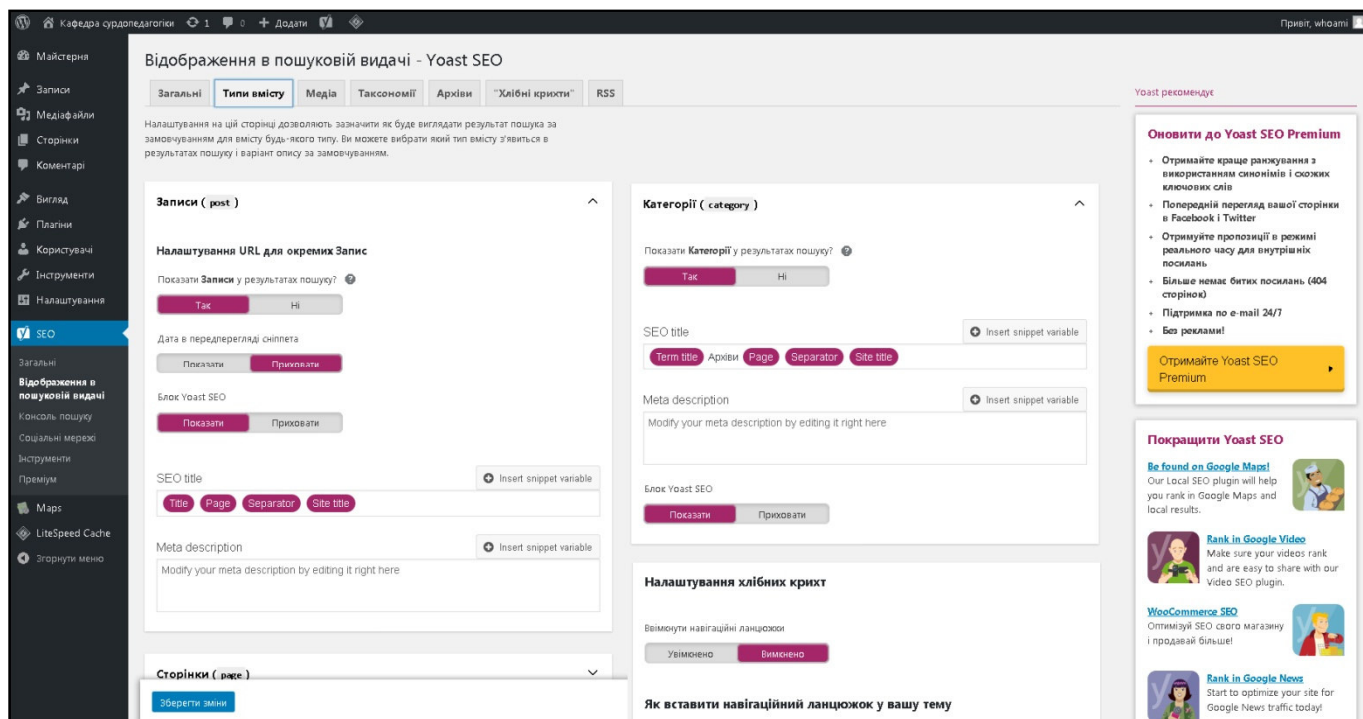


Рис. 3.24. Налаштування *SEO*-оптимізації

Фінальним кроком публікуємо весь контент у відповідності до рубрик веб-ресурсу. Вбудовуємо посилання в необхідних місцях на програвач відео від *YouTube* та шорткоди плагіну *RoboGallery* для відображення завантажених галерей фотофайлів.

3.3. Тестування веб-ресурсу

Починаємо тестування веб-ресурсу з ранжування його пошуковими системами: *Google*, *Yahoo*, *DuckDuckGo* (рис. 3.25). Відзначаємо, що веб-ресурс скрізь розташовується на першій сторінці. Констатуємо розміщення в усіх системах на першій позиції, окрім *Google* через відсутність *SSL*-сертифікату.

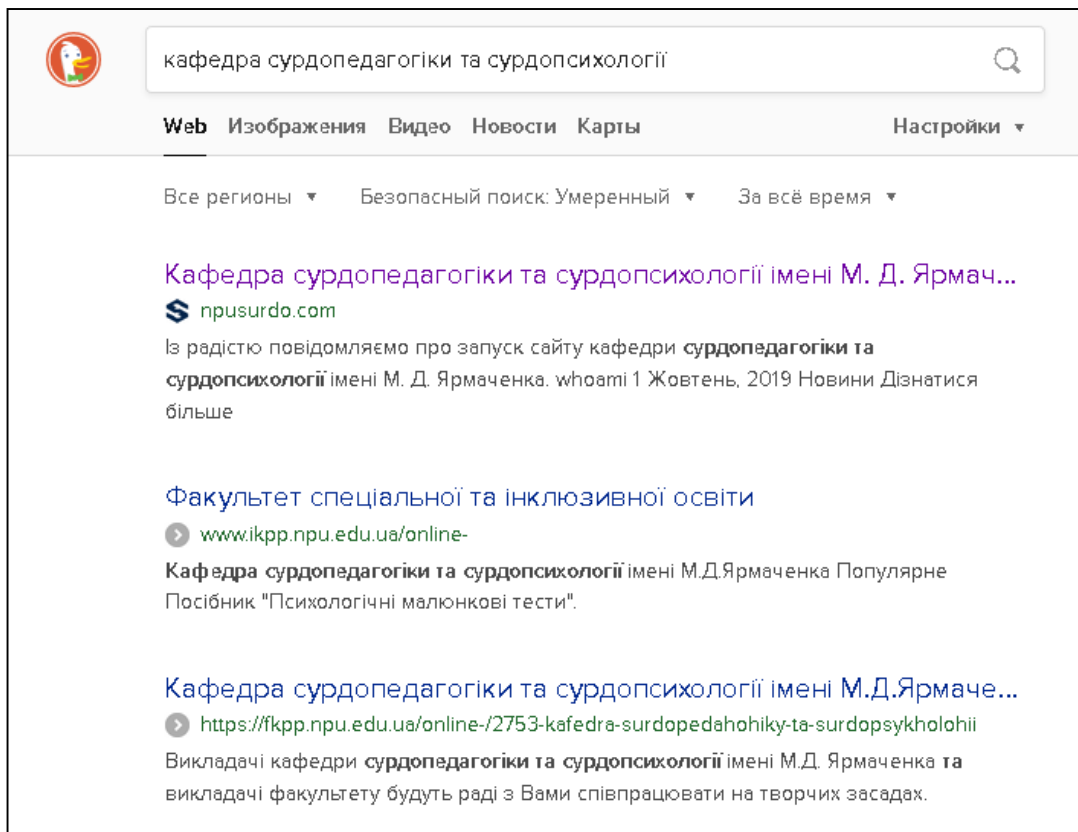


Рис. 3.25. Тестування в пошукових системах

Для перевірки кросплатформеності обираємо всі сучасні популярні браузері: *Mozilla Firefox*, *Google Chrome*, *Opera* та *Internet Explorer*. Отримані результати вказують на відсутність значних відмінностей в відображенні структури сторінок та дизайнерського оформлення (рис. 3.26).

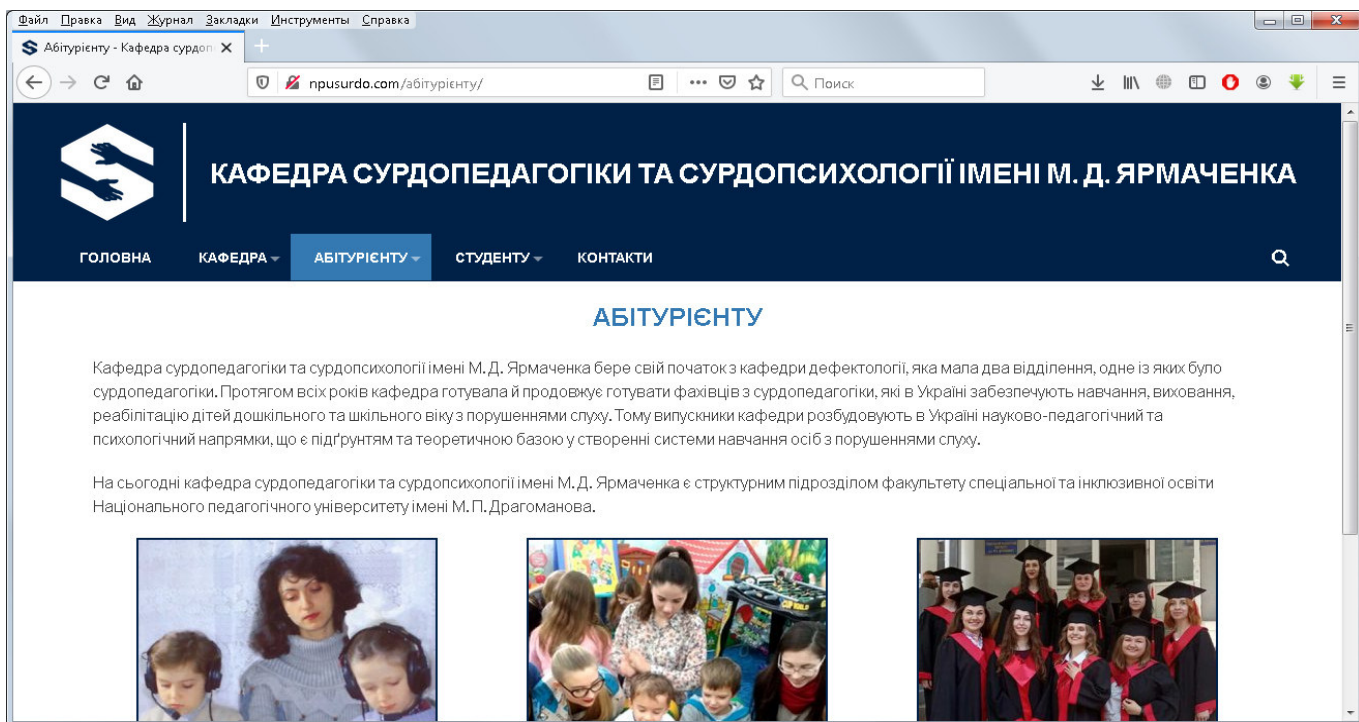


Рис. 3.26. Тестування кросплатформеності

Тестуємо швидкість завантаження веб-ресурсу за допомогою інструментів розробника, що вбудовано в веб-браузер. Перше завантаження показало результат 640 мілісекунд (рис. 3.27). А всі наступні спроби показали результат 400 мілісекунд, завдяки використанню системи кешування даних. Такі показники свідчать про відповідність нормам завантаження веб-ресурсів (2-3 секунди).

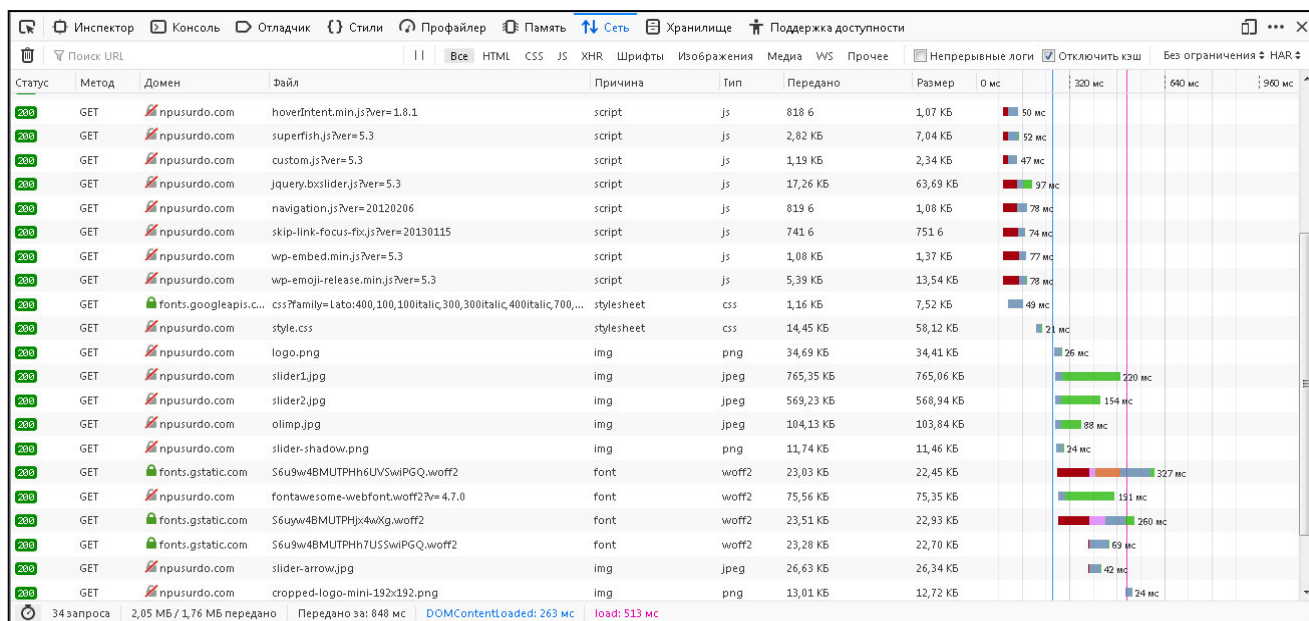


Рис. 3.27. Швидкість завантаження веб-ресурсу

Перевіряємо адаптивність хедеру, який є головним навігаційним елементом на всіх сторінках веб-ресурсу (рис. 3.28). При зміні розміру екрану констатуємо, що всі елементи (логотип, назва кафедри, меню навігації) змінюють місце розташування та зберігають логічну структурну побудову.

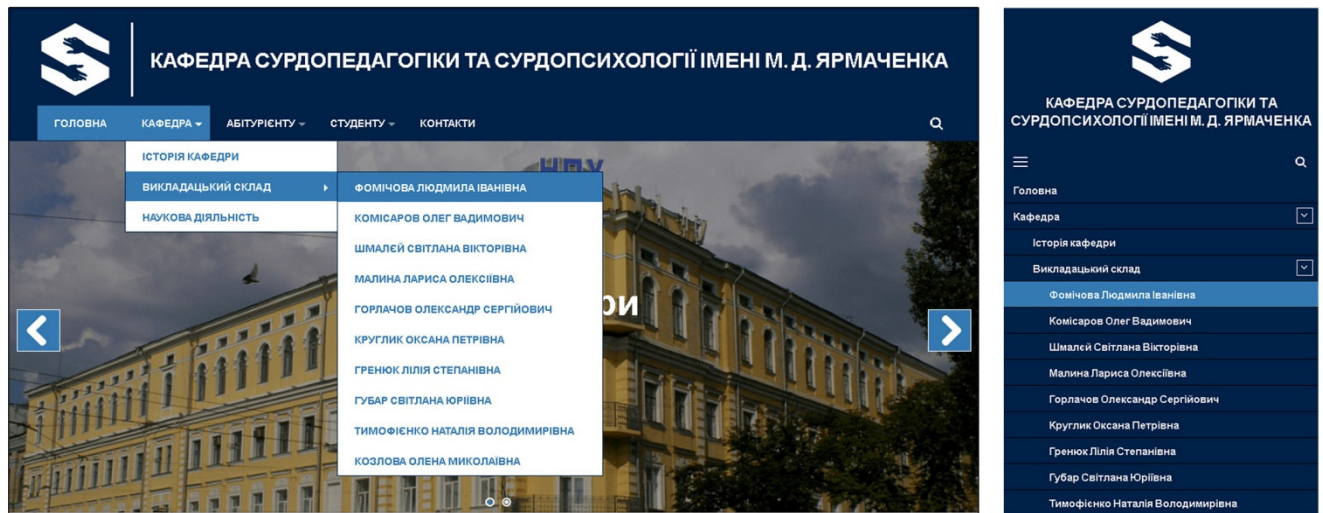


Рис. 3.28. Адаптивність хедеру

Змінюючи розміри екрану переглядаємо, як працює адаптивність на сторінці викладацького складу кафедри, що виконана за допомогою технології гнучкої верстки *Flexbox* (рис. 3.29).



Рис. 3.29. Адаптивність сторінки викладацького складу

Виконуємо тест на працездатність мапи та адаптивності сторінки з контактною інформацією. Зазначаємо, що мапа коректно працює (реагує на масштабування та швидко завантажується), а блоки з інформацією не змінюють структурного розташування (рис. 3.30).

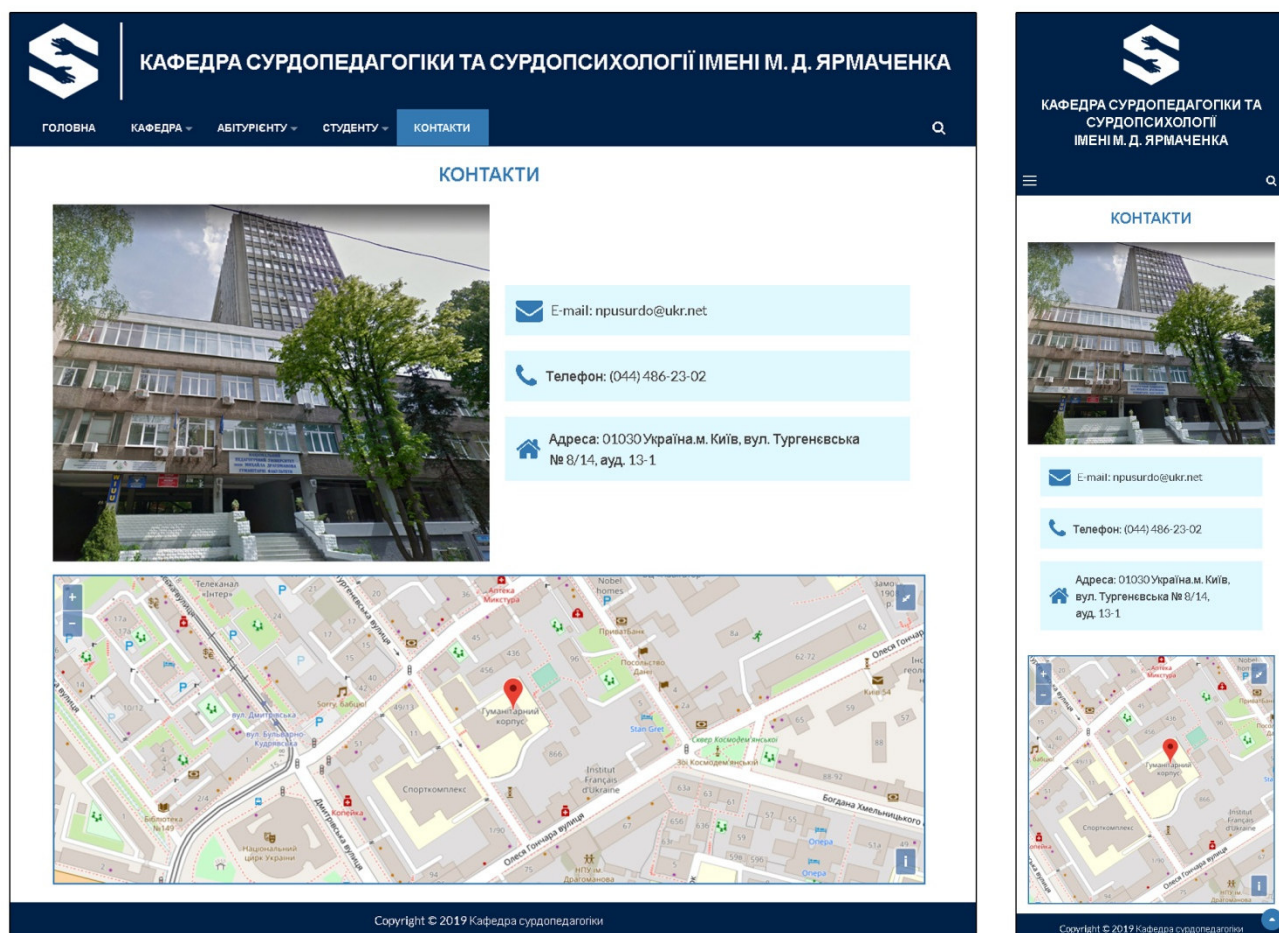


Рис. 3.30. Адаптивність сторінки контакти

Перевіряємо роботу пошукової системи веб-ресурсу. Спочатку вводимо інформацію, що завідомо відсутня, на яку система реагує повідомленням з можливістю подальшого пошуку (рис. 3.31). Потім створюємо запит на існуючий контент та отримуємо відповідні результати (рис. 3.32).

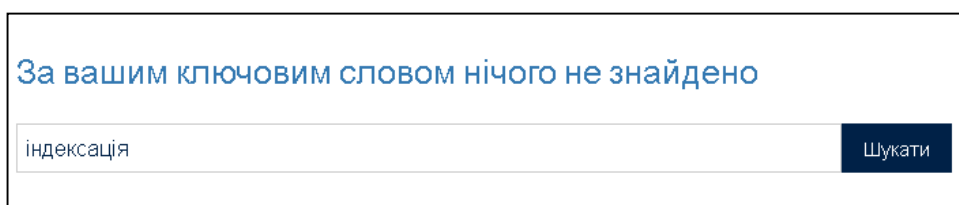


Рис .3.31. Тестування невірного пошуку

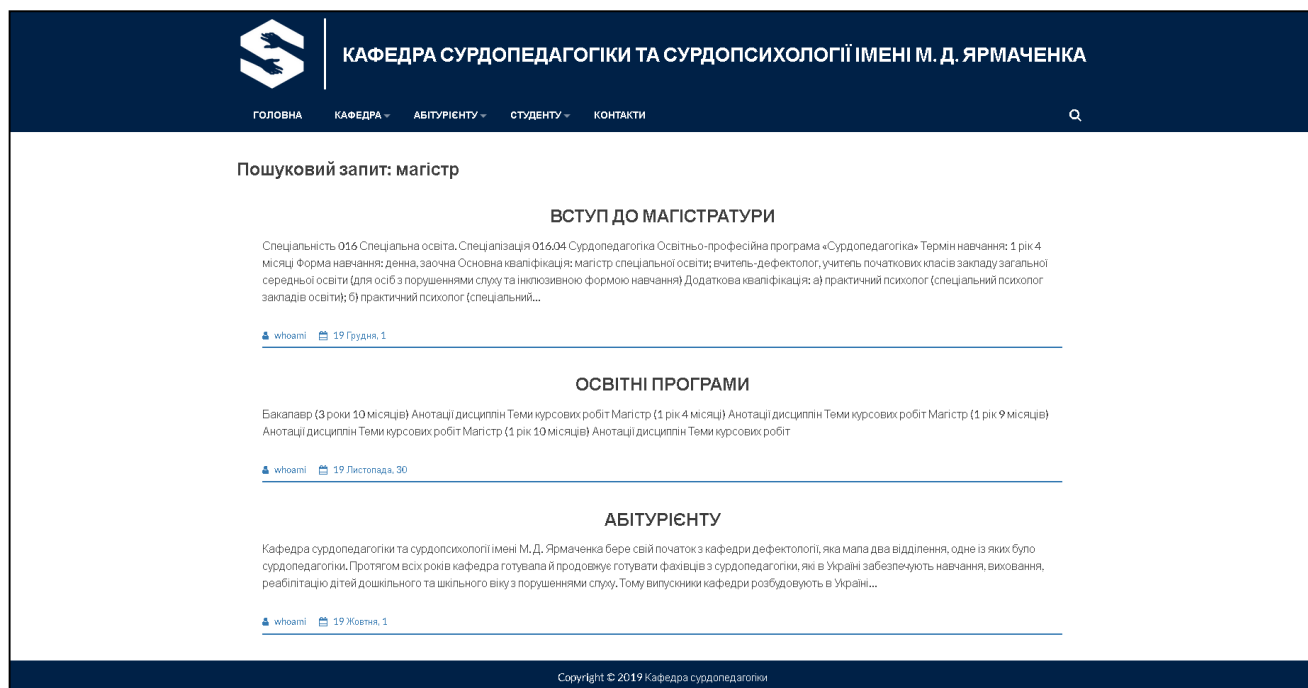


Рис. 3.32. Тестування пошуку

Здійснюємо тестовий контроль на наявність сторінки помилки під час введення користувачем невірної *URL*-адреси (рис. 3.33).

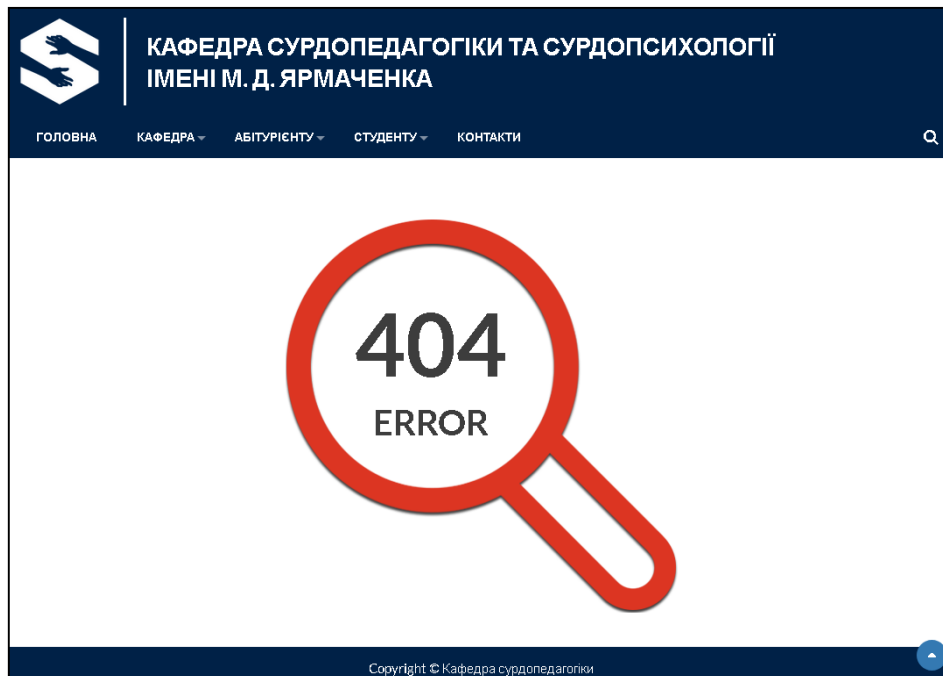


Рис. 3.33. Перевірка неіснуючих сторінок

Завершальним кроком перевіряємо всі взаємозв'язки між сторінками веб-ресурсу, а саме наявність всіх посилань та відсутність переходів.

Висновки до розділу

Розроблено внутрішню структуру, яка визначає логічний ієрархічний розподіл вмісту на структурні одиниці та взаємозв'язки між ними. Основу дизайнерської концепції складає кольорова комбінація (темно-синього, блакитного та білого), прямокутна форма елементів та шрифт *Lato*, що входить до сімейства без засічок. Відповідно до фірмового стилю розроблено унікальний логотип, що слугує безпомилковій зоровій ідентифікації кафедри в мережі Інтернет. Розташування ілюстративного та текстового матеріалу на сторінках веб-ресурсу виконано з дотриманням принципів дизайну та спроектовано варіанти їх адаптивного відображення.

Верстку та програмування веб-ресурсу здійснено в системі керування вмістом *WordPress* в якій обрано базовий шаблон, на основі якого створено дочірню тему. Виконано модернізацію *CMS* шляхом інсталяції функціональних плагінів з подальшим їх налаштуванням. За допомогою візуального редактору сформовано структуру меню з усіма рівнями вкладеності та описано зовнішній вигляд мовою стилів. Застосовуємо *CSS* для додавання декоративного оформлення із використанням псевдо-елементів та псевдо-класів. Інтерактивні можливості веб-ресурсу реалізовано з залученням бібліотеки *jQuery* (конструктивні особливості слайдеру, анімації та списки з прихованим вмістом). Мовою *PHP* запрограмовано функціональні елементи та їх відображення (панелі керування, рядки інформації), також прописано шорткоди для швидкого оформлення фрагментів вмісту. Застосовано технологію *Flexbox* при розробці варіантів адаптивного відображення на пристроях з різними розмірами екранів.

Здійснено тестування веб-ресурсу за наступними критеріями: ранжування пошуковими системами, кросплатформеність, швидкість завантаження, адаптивність, пошук по ресурсу та наявність сторінки помилки. Проведене тестування вказує, що даний веб-ресурс відповідає інформаційно-концептуальній моделі, створений у відповідності до норм семантики та *SEO*-оптимізації, підтримується всіма сучасними браузером та адаптується під будь-які розміри.

ВИСНОВКИ

Виконана дипломна робота дала змогу сформулювати наступні висновки відповідно до поставлених завдань.

1. На основі аналізу наукових праць досліджено видо-типологічну класифікацію та принципи проектування веб-ресурсів.

За видовими ознаками веб-ресурси розподіляються на: комерційні, корпоративні та інформаційні ресурси. За типологічними ознаками веб-ресурси розподіляються: залежно від мети створення, за ступенем доступу, за функціональністю та стилем оформлення, за фізичним розташуванням та за функціональною спрямованістю.

Принципи проектування веб-ресурсів базуються на трьох взаємопов'язаних складових: дизайні, програмуванні та *SEO*-оптимізації. Дизайн відповідає за візуальне відображення, що опирається на принципи: рівноваги, акцентування, прозорості, естетичності, ефективності, передбачуваності, орієнтованості на користувача, лаконічності та гармонії. Програмування реалізує функціональні можливості з використанням принципів: контрольованості, адаптивності, доступності, сумісності, реагування, багатомовності, персоналізації, збереження, корекційності та інформаційної підтримки. *SEO*-оптимізація сприяє підвищенню ранжування веб-ресурсу серед пошукових систем з дотриманням принципів: щільності ключових слів, побудови посилань, альтернативного тексту, оптимальної *URL*-адреси, прив'язки тексту, тегів заголовків та мета-описів.

В дипломній роботі виконано видо-типологічну класифікацію веб-ресурсу кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка. Згідно даної класифікації веб-ресурс кафедри за видом відноситься до інформаційного тематичного, а за типом – до некомерційного, відкритого, динамічного, глобального. В основу проектування веб-ресурсу кафедри покладено принципи: дизайну (естетичність, рівновага, акцентування, передбачуваність), програмування (адаптивність, реагування, сумісність, персоналізація) та *SEO*-оптимізації (щільність ключових слів, альтернативний текст, оптимальні *URL*-адреси, мета-описи).

2. Виконано порівняльний аналіз програмного забезпечення для написання коду та обрано текстовий редактор *Sublime Text 3*, враховуючи наступні основні критерії: персоналізація інтерфейсу, наявність підсвічування синтаксису, здатність розділення на робочі зони, можливість автоматичних відступів та автодоповнення коду, інтеграція модулів та додатків, відображення міні-мапи. Редактор вдосконалено функціональними плагінами (*sublime linter, emmet, colorPicker, advanced new file, liveReload, side bar enhancements, color highlighter, sublimeCodeIntel*) та написано сніппети для швидкого набору коду.

Здійснено порівняльний аналіз програмного забезпечення для створення дизайнерської концепції та обрано графічний редактор *CorelDRAW*, враховуючи наступні основні критерії: зручність інтерфейсу, функціональність інструментів, швидкість роботи, можливості роботи з текстом, налаштування експорту файлів, підтримка кольорових моделей, можливість роботи з популярними форматами.

Проведено порівняльний аналіз систем керування веб-ресурсів та обрано *CMS WordPress*, враховуючи наступні основні критерії: швидкодія, зручність користування, дизайнерські можливості, доступність покращення, можливості оптимізації пошукових систем та оновлюваність. Систему вдосконалено шляхом додавання плагінів (*Classic Editor, Smush, Yoast SEO, WP Google Maps, TinyMCE Advanced, Broken Link Checker, WordFence Security, Google Analytics*) та створено шорткоди для візуального виокремлення вмісту.

3. Визначено етапи технології створення веб-ресурсу „Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова”:

- формування технічного завдання (визначення мети створення та кінцевої цільової аудиторії, складання структурної схеми з вказанням взаємозв’язків між її елементами, опис шрифтового та кольорового оформлення, вказання необхідності підтримки можливості кросплатформеності та адаптивності);

- вибір хостингу (базується на основі оптимального співвідношення цінової політики і обсягу зберігаємої інформації) та доменного імені (відповідність назви до цілей веб-ресурсу);

- вибір *CMS* (ґрунтується на відповідності призначення веб-ресурсу, зручності користування та налаштування системи);
- розробка дизайнерської концепції (створення унікального візуального стилю з дотриманням принципів дизайну);
- верстка та програмування (реалізація дизайнерської концепції за допомогою мов *HTML, CSS, JavaScript, PHP, MySQL*);
- наповнення контентом (розміщення матеріалів на відповідних сторінках з використанням візуального редактору);
- *SEO*-оптимізація (застосування *SEO*-принципів для покращення ранжування в пошукових системах та збільшення швидкості завантаження);
- тестування (перевірка веб-ресурсу на відповідність всім нормам та відсутність помилок);
- публікація (надання можливості перегляду веб-ресурсу користувачам в мережі Інтернет).

Розроблену технологію створення веб-ресурсів можуть використовувати веб-розробники, як методичні рекомендації для створення ресурсів відповідного спрямування. Матеріали дослідження можуть бути застосовані в навчальному процесі в закладах, які займаються питаннями мультимедійної розробки.

4. В результаті виконання дипломної роботи розроблено технологію та створено веб-ресурс кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка Національного педагогічного університету імені М.П. Драгоманова, який відповідає всім нормам дизайну, програмування та *SEO*-оптимізації. Веб-ресурс виконує комунікативні функції, які полягають в наданні інформації про умови вступу та навчання, новини та анонси наукових заходів, викладацький склад, навчальні програми, розклад екзаменів та занять, контактну інформацію. Також він реалізує рекламну функцію для приваблення абітурієнтів перспективами майбутньої професії.

В графічному редакторі *CorelDRAW* розроблено внутрішню та зовнішню структури, обрано кольорову комбінацію (темно-синього, блакитного та білого), підібрано шрифт *Lato* для оформлення вмісту веб-ресурсу, створено логотип

кафедри, спроектовано сітку та виконано позиціонування елементів на сторінці, сконструйовано адаптивні варіанти відображення сторінок. Результатом є інформаційно-концептуальна модель веб-ресурсу, що повністю відповідає дизайнерським принципам проектування.

За допомогою текстового редактору *Sublime Text 3* в системі керування вмістом *WordPress* здійснено верстку та програмування веб-ресурсу кафедри. Використано мови: *HTML* для розмітки структурних семантичних блоків, *CSS* для написання стилів оформлення веб-ресурсу, *JavaScript* для додавання інтерактивних можливостей, *PHP* для генерації вмісту та програмування функціональних елементів, *MySQL* для створення запитів до бази даних. Додатково залучено бібліотеку *jQuery* для пришвидшення написання *JavaScript*-коду та застосовано технологію *Flexbox* для гнучкої адаптивної верстки.

Проведено тестування, що засвідчило про відповідність веб-ресурсу нормам семантичної верстки та *SEO*-оптимізації, підтримці кросплатформеності та адаптивності, присутності сторінки помилки, справному пошуку по ресурсу, наявності високої швидкості завантаження завдяки оптимізації файлів.

Виконане дослідження не вичерпує всіх аспектів означеної проблеми, подальшого дослідження потребують питання розробки системи керування вмістом та програмування плагінів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Березко О.Л., Пелешишин А.М., Жежнич П.І. Концепція створення веб-сайта Національного університету „Львівська політехніка” / О.Л. Березко, А.М. Пелешишин, П.І. Жежнич [Електронний ресурс]. – Режим доступу: <http://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/3146/12-berezko-57-65.pdf> (дата звернення 19.12.2019).
2. Берковський В.В., Радіонов В.О., Бурковський В.С. Зіставлення технологій створення сайтів з їх цільовим призначенням / В.В. Берковський, В.О. Радіонов, В.С. Бурковський // Наука і техніка Повітряних Сил Збройних Сил України. – 2015. – № 2(19). – С. 122-124.
3. Браэм Г. Психология цвета / Г. Браэм. – Москва: АСТ, 2009. – 158 с.
4. Васильев А.Н. JavaScript в примерах и задачах / А.Н. Васильев. – Москва: Издательство „Э”, 2017. – 720 с.
5. Гушев А.А. Классификация web-сайтов образовательных учреждений / А.А. Гушев // Современные научные исследования и инновации. – 2016. – № 11. [Електронний ресурс]. – Режим доступу: <http://web.snauka.ru/issues/2016/11/74309> (дата звернення 19.12.2019).
6. Дакетт Д. JavaScripte и jQuery. Интерактивная веб-разработка / Д. Дакетт. – Москва: Издательство „Э”, 2017. – 640 с.
7. Евдокимов Н.С. Создание сайтов / Н.С. Евдокимов. – Санкт-Петербург: Питер, 2014. – 410 с.
8. Компанєєтс М.О. Принципи проектування ефективних веб-сайтів / М.О. Компанєєтс // Молодий вчений. – 2015. – № 9. – Част. 2. – С. 106-108. [Електронний ресурс]. – Режим доступу: <http://molodyvcheny.in.ua/files/journal/2015/9/66.pdf> (дата звернення 19.12.2019).
9. Котеров Д.В. PHP 7 / Д.В. Котеров. – Санкт-Петербург: БХВ-Петербург, 2016. – 1088 с.
10. Круг С. Веб-дизайн: книга Стива Круга или „не заставляйте меня думать!” / С. Круг. – 2-е изд. – Санкт-Петербург: Символ-Плюс, 2008. – 224 с.

11. Лещев А.В. Создание интерактивного web-сайта : учебный курс / А.В. Лещев. – Санкт-Петербург: Питер, 2003. – 544 с.
12. Мак-Дональд М. HTML5. Недостающее руководство / М. Мак-Дональд. – Санкт-Петербург: БХВ-Петербург, 2012. – 480 с.
13. Мак-Дональд М. Создание Web-сайта. Недостающее руководство / М. Мак-Дональд. – 3-е изд. – Санкт-Петербург: БХВ-Петербург, 2013. – 624 с.
14. Макфарланд Д. Новая большая книга CSS / Д. Макфарланд. – Санкт-Петербург: Питер, 2016. – 720 с.
15. Мальцев И.П. Проектирование веб-сайта: пошаговое руководство для тех, кто понял, что сайт ему необходим, но пока не знает, что с этим делать... / И.П. Мальцев. – ЛитРес: Самиздат, 2018. – 118 с. [Электронный ресурс]. – Режим доступа: <https://prowebber.ru/text-lessons/20608-kniga-proektirovanie-veb-sayta-imalcev.html> (дата звернения 19.12.2019).
16. Мельник О. Выбираем профессию frontend- и backend-разработчика: принципы и отличия / О. Мельник [Электронный ресурс]. – Режим доступа: https://skillbox.ru/media/code/frontend_i_backend_razrabotka/ (дата звернения 19.12.2019).
17. Мейер Э., Уэйл Э. CSS: полный справочник / Э. Мейер, Э. Уэйл. – Санкт-Петербург: ООО „Диалектика”, 2019. – 1088 с.
18. Милин В. Обзор редакторов кода / В. Милин [Электронный ресурс]. – Режим доступа: <https://htmlacademy.ru/blog/useful/programming/editors-for-the-coders> (дата звернения 19.12.2019).
19. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – Санкт-Петербург: Питер, 2016. – 768 с.
20. Пасічник Н.Р. Формалізм в постановці задачі створення якісного сайту / Н.Р. Пасічник // Наукові праці Донецького національного технічного університету. Інформатика, кібернетика та обчислювальна техніка. – Донецьк. – 2011. – Вип. 14 (188). – С. 325-329. [Електронний ресурс]. – Режим доступу: http://nbuv.gov.ua/UJRN/Npdntu_inf_2011_14_49 (дата звернення 19.12.2019).

21. Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В. Інформатика: підручник для 11 кл. загальноосвіт. навч. закл. : академ. рівень, профільн. рівень / Й.Я. Ривкінд, Т.І. Лисенко, Л.А. Чернікова, В.В. Шакотько / за заг. ред. М.З. Згуровського. – Київ: Генеза, 2011. – 304 с.
22. Робин Н., Греди К. Эффективный Web-сайт : учебн. пос. / Н. Робин, К. Греди. – Москва: Триумф, 2004. – 560 с.
23. Самойлова Т.И. Основные виды сайтов / Т.И. Самойлова [Электронный ресурс]. – Режим доступа: <https://tutor-web.susu.ru/2017/07/12/osnovnyie-vidyi-saytov/> (дата звернення 15.10.2019).
24. Самойлова Т.И. Основные типы сайтов / Т.И. Самойлова [Электронный ресурс]. – Режим доступа: <https://tutor-web.susu.ru/2017/07/12/osnovnyie-tipyi-saytov/> (дата звернення: 15.10.2019).
25. Томал Р. Основы Web-Дизайна. Руководство / Р. Томал [Электронный ресурс]. – Режим доступа: <http://padabum.com/d.php?id=191574> (дата звернення 19.12.2019).
26. Фримен Э., Робсон Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Санкт-Петербург: Питер, 2015. – 640 с.
27. Шевченко Д.А., Локтюшина Ю.В. Эффективность веб-сайтов высших учебных заведений. Методика оценки конкурентоспособности сайта вуза в Интернет / Д.А. Шевченко, Ю.В. Локтюшина. – Москва: МИПК, 2014. – 141 с.
28. Энж Э., Спенсер С., Стрикчиола Д. SEO – искусство раскрутки сайтов / Э. Энж, С. Спенсе, Д. Стрикчиола. – 3-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2017. – 816 с.
29. Atom Flight Manual [Electronic resource]. – Access mode: <https://flight-manual.atom.io/> (last access 19.12.2019).
30. Barraclough D. What are the different types of websites? / D. Barraclough [Electronic resource]. – Access mode: <https://www.expertmarket.co.uk/web-design/different-types-of-websites> (last access 15.10.2019).
31. Beaird J. The Principles of Beautiful Web Design / J. Beaird. – Canada: SitePoint Pty. Ltd, 2010. – 218 p. [Electronic resource]. – Access mode:

<https://doc.lagout.org/programmation/Design/The%20Principles%20of%20Beautiful%20Web%20Design%20-%20Beaird%20-%20Site%20Point%20%282010%29%20.pdf> (last access 19.12.2019).

32. Bhaskar N.U., Prathap N.P., Chandra R.B., Govindarajulu P. General Principles of User Interface Design and Websites / N.U. Bhaskar, N.P. Prathap, R.B. Chandra, P. Govindarajulu // International Journal of Software Engineering (IJSE). – 2011. – Vol. 2. – Iss. 3. – P. 45-60. [Electronic resource]. – Access mode: <https://www.cscjournals.org/manuscript/Journals/IJSE/Volume2/Issue3/IJSE-54.pdf> (last access 19.12.2019).

33. Boldt T. The Best Vector Graphics Software / T. Boldt [Electronic resource]. – Access mode: <https://www.softwarehow.com/best-vector-graphics-software/> (last access 19.12.2019).

34. Booth D., Jansen B.J. A Review of Methodologies for Analyzing Websites / D. Booth, B.J. Jansen. – Pennsylvania State University, USA: IGI Global. – 2009. – P. 141-162. [Electronic resource]. – Access mode: https://faculty.ist.psu.edu/jjansen/academic/jansen_website_analysis.pdf (last access 19.12.2019).

35. Burgstahler S. Universal Design: Process, Principles, and Applications / S. Burgstahler [Electronic resource]. – Access mode: https://www.washington.edu/doit/sites/default/files/atoms/files/Universal_Design%20Process%20Principles%20and%20Applications.pdf (last access 19.12.2019).

36. Costa C.J., Costa P., Aparício M. Principles for Creating Web Sites / C.J. Costa, P. Costa, M. Aparício // A Design Perspective: Conference Paper, January 2004. [Electronic resource]. – Access mode: https://www.researchgate.net/publication/220709794_Principles_for_Creating_Web_Sites_A_Design_Perspective (last access 19.12.2019).

37. Chandra S. Different Types of Websites / S. Chandra [Electronic resource]. – Access mode: <https://deepknowhow.com/2014/06/02/different-types-websites/> (last access 19.12.2019).

38. Chapman C. Color Theory for Designers, Part 1: The Meaning of Color / C. Chapman [Electronic resource]. – Access mode: <https://www.smashingmagazine.com/2010/01/color-theory-for-designers-part-1-the-meaning-of-color/> (last access 19.12.2019).
39. Documentation Drupal [Electronic resource]. – Access mode: <https://www.drupal.org/documentation> (last access 19.12.2019).
40. Documentation Sublime Text [Electronic resource]. – Access mode: <https://www.sublimetext.com/docs/3/> (last access 19.12.2019).
41. Ewer T. 30 Essential WordPress Plugins You Should Install If You Haven't Already / T. Ewer [Electronic resource]. – Access mode: <https://premium.wpmudev.org/blog/essential-wordpress-plugins/> (last access 19.12.2019).
42. Garcia S. IDE? Text Editor? What's the difference? / S. Garcia [Electronic resource]. – Access mode: <https://medium.com/@sgarcia.dev/learn-webdev-series-coding-tools-part-1-introduction-ides-vs-text-editors-and-the-rise-of-the-75c74bf26988> (last access 19.12.2019).
43. Getting Started Visual Studio Code [Electronic resource]. – Access mode: <https://code.visualstudio.com/docs> (last access 19.12.2019).
44. Graham N. 8 Different Types of Websites / N. Graham [Electronic resource]. – Access mode: <https://ezinearticles.com/?8-Different-Types-of-Websites&id=4340854> (last access 12.2019).
45. ISBD(ER): International Standart Bibliographic Description for Electronic Resources [Electronic resource]. – Access mode: <https://www.ifla.org/files/assets/cataloguing/isbd/isbder.pdf> (last access 15.10.2019).
46. Joomla! Documentation. [Electronic resource]. – Access mode: https://docs.joomla.org/Main_Page (last access 19.12.2019).
47. Kipp M.E.I., Joo S., Choi I. Classification of Web Resources using User Generated Terms / M.E.I. Kipp, S. Joo, I. Choi [Electronic resource]. – Access mode: <http://library.ifla.org/186/1/112-kipp-en.pdf> (last access 19.12.2019).
48. Leavitt M.O., Shneiderman B. Research-Based Web Design & Usability Guidelines / M.O. Leavitt, B. Shneiderman. – Washington: Department of Health and Human Services, 2006. – 267 p.

49. Liew Z. Gulp for Beginners / Z. Liew [Electronic resource]. – Access mode: <https://css-tricks.com/gulp-for-beginners/> (last access 19.12.2019).
50. Lindley C. Front-end Developer Handbook 2019 / C. Lindley. [Electronic resource]. – Access mode: <https://frontendmasters.com/books/front-end-handbook/2019/> (last access 19.12.2019).
51. Markov D. 15 Awesome Sublime Text Plugins For Web Development / D. Markov [Electronic resource]. – Access mode: <https://tutorialzine.com/2016/10/15-awesome-sublime-text-plugins-for-web-development> (last access 19.12.2019).
52. Matveyeva S.J. A Role for Classification: The Organization of Resources on the Internet / S.J. Matveyeva [Electronic resource]. – Access mode: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.534.8427&rep=rep1&type=pdf> (last access 19.12.2019).
53. Mehta N. Choosing an Open Source CMS. Beginner's Guide / N. Mehta. – Birmingham-Mumbai: Packt Publishing, 2009. – 318 p.
54. Meyer zu Eissen S., Stein B. Genre Classification of Web Pages / S. Meyer zu Eissen, B. Stein. – Paderborn, Germany [Electronic resource]. – Access mode: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.2987&rep=rep1&type=pdf> (last access 15.10.2019).
55. Miletsky J.I. Principles of Internet Marketing: New Tools and Methods for Web Developers : Course Technology Press / J.I. Miletsky. – Boston, MA, United States, 2009. – 480 p.
56. Moore S. Core Principles of SEO / S. Moore [Electronic resource]. – Access mode: <https://www.websitemagazine.com/blog/core-principles-of-seo> (last access 19.12.2019).
57. Morris W. 8 Best CMS Platforms to Start a Website in 2020 / W. Morris [Electronic resource]. – Access mode: <https://www.hostinger.com/tutorials/best-cms> (last access 19.12.2019).
58. MySQL Documentation [Electronic resource]. – Access mode: <https://dev.mysql.com/doc/> (last access 19.12.2019).

59. Principles of Mobile App Design: Engage Users and Drive Conversions [Electronic resource]. – Access mode: <https://www.thinkwithgoogle.com/marketing-resources/experience-design/principles-of-mobile-app-design-engage-users-and-drive-conversions/> (last access 19.12.2019).

60. Quality Principles for Cultural Websites: a Handbook. – Rome: La Tipografia, Italy, 2005. – 66 p. [Electronic resource]. – Access mode: <http://www.minervaeurope.org/publications/qualitycommentary/qualitycommentary050314final.pdf> (last access 19.12.2019).

61. Rajalakshmi R., Aravindan C. Naive Bayes Approach for Website Classification / R. Rajalakshmi, C. Aravindan // Information Technology and Mobile Communication: International Conference, AIM 2011, Nagpur, Maharashtra, India, April 21-22, 2011. – P. 323-326. [Electronic resource]. – Access mode: https://www.researchgate.net/publication/225983336_Naive_Bayes_Approach_for_Website_Classification (last access 19.12.2019).


62. Roussinov D., Crowston K., Nilan M., Kwasnik B., Cai J., Liu X. Genre Based Navigation on the Web / D. Roussinov, K. Crowston, M. Nilan, B. Kwasnik, J. Cai, X. Liu // Hawaii International Conference on System Sciences (HICSS-34). – Island of Maui, January 4-7, 2001 [Electronic resource]. – Access mode: <https://surface.syr.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1117&context=istpub> (last access 15.10.2019).

63. Satija M.P., Martinez-Avila D. Use of classification in organizing and searching the web / M.P. Satija, D. Martinez-Avila // Annals of Library and Information Studies. – 2014. – Vol. 61. – P. 294-306. [Electronic resource]. – Access mode: <http://nopr.niscair.res.in/bitstream/123456789/30337/4/ALIS%2061%284%29%20294-306.pdf> (last access 19.12.2019).

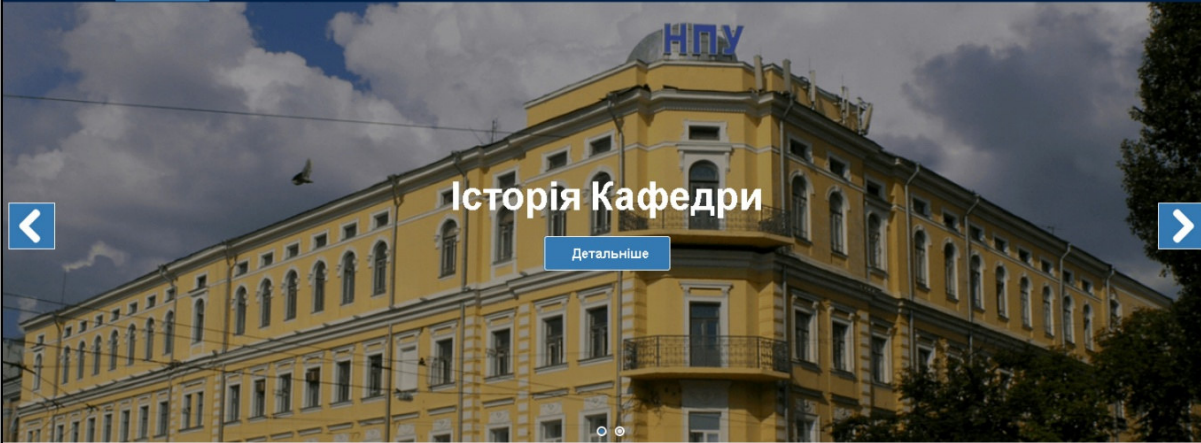
64. Sahu P. SEO for web designers: What you should know / P. Sahu [Electronic resource]. – Access mode: <https://www.searchenginewatch.com/2018/10/23/seo-for-web-designers-what-you-should-know/> (last access 19.12.2019).

65. Schäferhoff N. 12 „Best” WordPress Plugins (Free & Paid) / N. Schäferhoff [Electronic resource]. – Access mode: <https://websitesetup.org/best-popular-wordpress-plugins/> (last access 19.12.2019).
66. Sitbon B.J. Step-By-Step Guide: How to Make a Professional Website in 2019 / B.J. Sitbon [Electronic resource]. – Access mode: <https://www.wix.com/blog/2019/03/how-to-make-website-guide> (last access 19.12.2019).
67. Slavic A. Classification revisited: a web of knowledge / A. Slavic // Innovations in information retrieval: perspectives for theory and practice. – London: Facet, 2011. – P. 23-48. [Electronic resource]. – Access mode: <https://arxiv.org/ftp/arxiv/papers/1705/1705.07058.pdf> (last access 19.12.2019).
68. Vasile C. Choosing the best code editor as a web developer in 2019 / C. Vasile [Electronic resource]. – Access mode: <https://designrevision.com/best-code-editor/> (last access 19.12.2019).
69. Winkler M. The A to Z of Adobe Illustrator / M. Winkler [Electronic resource]. – Access mode: <https://design.tutsplus.com/articles/the-a-to-z-of-adobe-illustrator--vector-20799> (last access 19.12.2019).
70. Winkler M. The A to Z of CorelDRAW / M. Winkler [Electronic resource]. – Access mode: <https://design.tutsplus.com/articles/the-a-to-z-of-coreldraw--cms-24194> (last access 19.12.2019).
71. Wodehouse C. A Beginner's Guide to Back-End Development / C. Wodehouse [Electronic resource]. – Access mode: <https://www.upwork.com/hiring/development/a-beginners-guide-to-back-end-development> (last access 19.12.2019).
72. WordPress Features [Electronic resource]. – Access mode: <https://wordpress.org/support/article/wordpress-features/> (last access 19.12.2019).
73. Zheng J.G., Peltsverger S. Web Analytics Overview / J.G. Zheng, S. Peltsverger // Encyclopedia of Information Science and Technology. – Third Edition. – Chapter 756. – Pennsylvania: IGI Global, 2015. [Electronic resource]. – Access mode: https://www.researchgate.net/publication/272815693_Web_Analytics_Overview (last access 19.12.2019).


Головна сторінка веб-ресурсу кафедри


КАФЕДРА СУРДОПЕДАГОГІКИ ТА СУРДОПСИХОЛОГІЇ ІМЕНІ М. Д. ЯРМАЧЕНКА




[ГОЛОВНА](#)
[КАФЕДРА](#)
[АВІТУРІЕНТУ](#)
[СТУДЕНТУ](#)
[КОНТАКТИ](#)




Всеукраїнська науково-практична конференція «Педагогічні читання — 2019». 60 років кафедрі сурдопедагогіки та сурдопсихології імені М. Д. Ярмаченка






Кафедра сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка факультету спеціальної та інклюзивної освіти Національного педагогічного університету імені М.П. Драгоманова, Громадська організація «Асоціація сурдопедагогів та піклувальників про осіб з вадами слуху» «Інститут сурдопедагогічної політики» проводила XXI Всеукраїнську науково-практичну конференцію «Педагогічні читання – 2019». Робота конференції складалась із двох напрямків: 29.10.2019 – пленарне засідання та круглий стіл з проблемами «Завдання...

 whoami
  9 Грудня, 2019
  Новини
 [Детальніше](#)


В жовтні 2019 року відбулась олімпіада на тему: «Сурдопедагог – моє покликання і виклики сучасності»






В жовтні 2019 року серед студентів – майбутніх сурдопедагогів була проведена олімпіада на тему: «Сурдопедагог – моє покликання і виклики сучасності». В олімпіаді взяли участь студенти всіх курсів. 30 жовтня було визначено переможців олімпіади. Переможці олімпіади були нагороджені грамотами, які вручила завідувач кафедри сурдопедагогіки та сурдопсихології імені М.Д. Ярмаченка доктор психологічних наук, професор Фомічова...

 whoami
  9 Грудня, 2019
  Новини
 [Детальніше](#)

Запуск




Із радістю повідомляємо про запуск сайту кафедри сурдопедагогіки та сурдопсихології імені М. Д. Ярмаченка.

 whoami
  1 Жовтня, 2019
  Новини
 [Детальніше](#)

Copyright © 2020 Кафедра сурдопедагогіки

Адаптивний вигляд сторінки



КАФЕДРА СУРДОПЕДАГОГІКИ ТА
СУРДОПСИХОЛОГІЇ ІМЕНІ М. Д. ЯРМАЧЕНКА

☰
🔍

Головна

Кафедра >

Абітурієнту >

Студенту ▾


Освітні програми


Підсумкова атестація

Творчість студентів


Контакти


ОСВІТНІ ПРОГРАМИ







Бакалавр
(3 роки 10 місяців)


Анотації дисциплін 


Теми курсових робіт 




Магістр
(1 рік 4 місяці)


Анотації дисциплін 


Теми курсових робіт 




Магістр
(1 рік 9 місяців)


Анотації дисциплін 

Теми курсових робіт 



Магістр
(1 рік 10 місяців)

Анотації дисциплін 

Теми курсових робіт 

Copyright © 2020 Кафедра сурдопедагогіки

⬆